

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE CANTABRIA**



***Trabajo Fin de Máster***

**Contribución al proceso de  
reacondicionamiento de amplificadores  
klystron para aceleradores lineales de  
partículas en ESS Bilbao**

**(Contribution to klystron amplifiers reconditioning  
process for linear particle accelerators  
at ESS Bilbao)**

**Para acceder al Título de**

**Máster Universitario en Ingeniería de Telecomunicación**

**Autor: Álvaro Guitián Herrera**

**Julio, 2020**

## Tabla de contenido

Resumen .....	8
Abstract .....	9
Capítulo 1: Introducción .....	10
1.1 Elementos básicos de un acelerador lineal de partículas .....	13
1.2 Esquema del acelerador de protones de la fuente ESS .....	16
1.3 Ejemplos de aplicaciones industriales y científicas de los aceleradores de partículas.....	18
Capítulo 2: Sistemas de radiofrecuencia para aceleradores .....	22
2.1 Klystron .....	25
2.2 RF test stand .....	29
Capítulo 3: Mejoras introducidas en los sistemas de enclavamientos rápidos.....	33
3.1 Sistema de enclavamientos rápidos de ESS Bilbao. ....	33
3.2 Comparadores de umbral del sistema fast interlocks .....	39
3.2.1 Mejoras introducidas en el comparador de umbral.....	41
3.3 Comparador de potencia del sistema Fast Interlocks.....	43
3.3.1 Sustitución del amplificador de instrumentación AD620.....	50
Capítulo 4: Interfaz gráfica en MATLAB para adquisición de medidas automáticas del klystron .....	55
4.1 Interfaz gráfica de usuario en MATLAB para medidas con el klystron.....	56
4.2 Estructura y código del GUI .....	65
4.3 Fichero de configuración.....	66
4.4 Funciones de retrollamada (callback) de los objetos del GUI .....	69
Capítulo 5: Medida de alta potencia por calorimetría.....	74
5.1 Técnicas de calorimetría para medida de alta potencia de RF.....	74
5.2 Interfaz gráfica de usuario para medidas de potencia con calorimetría con el software LabView .....	83

5.3 Cálculos de la medida de potencia de RF por calorimetría .....	89
5.4 Causas y fuentes de error en la medida .....	90
Capítulo 6: Conclusiones y líneas futuras .....	93
REFERENCIAS.....	96
ANEXO I: Código fuente de la GUI para medidas del Klystron.....	98
ANEXO II: Procesado del fichero de resultados de la GUI de MATLAB .....	119

## Indice de figuras y tablas

Figura 1.1 Esquema de un LINAC simplificado.....	14
Tabla 1.1 Parámetros del modulador de ESS Bilbao .....	15
Figura 1.2 Secciones del acelerador de protones de ESS y estaciones de RF para la sección no superconductora. ....	16
Figura 1.3 Esquema que muestra los componentes básicos y el funcionamiento de los sistemas de procesamiento de materiales.....	18
Figura 1.4 Acelerador lineal para tratamiento médico.....	20
Figura 2.1 Sistema de radiofrecuencia típico .....	24
Figura 2.2 Estructura básica de un klystron de tres cavidades [17] .....	26
Figura 2.3 Klystron de CPI, ESS Bilbao .....	26
Figura 2.4 Klystron con el blindaje incorporado, ESS Bilbao. ....	27
Figura 2.5 Entrada de alta tensión negativa en el klystron .....	28
Figura 2.6 Test Stand de RF de alta potencia en Zamudio .....	31
Figura 2.7 Klystron instalado en ESS ERIC (Lund, Suecia).....	32
Figura 3.1 Frontal del sistema fast interlocks .....	36
Figura 3.2 Circuito del comparador de umbral del sistema fast interlocks .....	36
Figura 3.3 Circuito del comparador de umbral del sistema fast interlocks .....	37
Figura 3.4 Detectores de arco, controladores de las bombas iónicas de vacío y fuente de alimentación del filamento.....	38
Figura 3.5 Pruebas de validación en ESS Bilbao del sistema detector de arcos para el RFQ de ESS ERIC.....	38
Figura 3.6 Comparador de umbral del sistema de interlocks rápidos.....	40
Figura 3.7 Esquemático en LTSpice del comparador con umbral negativo .....	42
Figura 3.8 Simulación del esquema comparador de umbral negativo .....	42
Figura 3.9 Voltaje de salida en función de la potencia de entrada al detector.....	44
Tabla 3.1 Valores de potencia en el detector y tensiones a la salida.....	44

Tabla 3.2 Valor del umbral en función de la ruleta del potenciómetro multi-vuelta .....	46
Figura 3.10 Diagrama temporal con el amplificador de instrumentación.....	47
Figura 3.11 Representación de la falsa alarma en el osciloscopio.....	47
Figura 3.12 Esquemático del comparador de potencia .....	48
Figura 3.13 Señales obtenidas de la simulación en LTSpice con el falso error, empleando el amplificador de instrumentación AD620 .....	49
Figura 3.14 Diagrama temporal con el amplificador de instrumentación AMP02 ..	50
Figura 3.15 Esquema interno del amplificador de instrumentación AMP02 de Analog Devices.....	51
Figura 3.16 Esquemático del AMP02 realizado en LTSpice de Analog Devices ....	51
Figura 3.17 Símbolo del componente de librería del amplificador de instrumentación AMP02 .....	52
Figura 3.19 Señales obtenidas en LTSpice con el amplificador de instrumentación AMP02 .....	54
Figura 4.1 Apertura de un nuevo GUI .....	56
Figura 4.2 Entorno de diseño de los objetos.....	57
Figura 4.3 Vista del entorno de diseño de la GUI .....	58
Figura 4.4 Menú desplegable de ‘property inspector’ de los objetos .....	59
Figura 4.5 Lista de objetos en la GUI.....	60
Figura 4.6 Aspecto final de la GUI .....	61
Figura 4.7 Salvando resultados en archivo .csv .....	63
Figura 4.8 Ejemplo de imagen guardada de medida PinVsPout con extensión ‘.jpg’ .....	64
Figura 4.9 Ejemplo de imagen guardada de medida frecuencial.....	64
Figura 4.10 Estructura de la GUI para medidas automáticas con los klystron .....	66
Figura 4.11 Software de medidas automáticas en el osciloscopio de ESS Bilbao...	73
Figura 5.1 Colector del klystron.....	76
Figura 5.2 Llave de paso y manómetro.....	76

Figura 5.3 Intercambiador de calor.....	77
Figura 5.4 Vaso de expansión y medidores de calidad del agua (pH, O <sub>2</sub> , conductividad eléctrica [ $\mu$ S/cm], temperatura [ $^{\circ}$ C]).....	77
Figura 5.5 Distribuidor de circuitos de agua independientes, compuestos por dos tubos cada uno (de ida y de retorno).....	78
Figura 5.6 Esquema de las sondas de medida temperatura, presión y caudal del circuito de agua .....	79
Figura 5.7 Sensores de temperatura PT100, y de presión .....	79
Figura 5.8 Caudalímetro .....	80
Figura 5.9 Ejemplo de panel frontal.....	83
Figura 5.10 Diagrama de bloques del panel frontal del ejemplo.....	84
Figura 5.11 Panel frontal del programa para la medida de alta potencia mediante calorimetría.....	85
Figura 5.12 Diagrama de bloques del programa desarrollado.....	86
Figura 5.13. Software de calorimetría implementado en la sala de control del RF Test Stand .....	88
Tabla 5.1 Valores de error en la potencia de RF debidas al detector .....	91

# Agradecimientos

Este TFM está integrado dentro del Proyecto del Consorcio ESS Bilbao titulado “Transmisores de Radiofrecuencia para Aceleradores de partículas basados en Klystrons y estado Sólido que contribuyen al desarrollo tecnológico de la sociedad (TRACKS)”, cofinanciado con fondos FEDER de la Unión Europea.

No hubiera sido posible llevar a cabo este trabajo sin la oportunidad que se me concedió de estar contratado durante cuatro magníficos meses en las instalaciones de ESS Bilbao en Zamudio. Durante esos meses aprendí innumerables conocimientos dentro de muchas áreas diferentes, y crecí profesionalmente rodeado de gente muy profesional y muy humana. A todos los que hacen posible ESS Bilbao, en especial a mi director del proyecto Pedro González, siempre con mucha disposición y de un trato muy cercano. Muchas gracias.

Por otro lado, quiero hacer una mención y agradecerle a Tomás Fernández, otro pilar fundamental de este trabajo y dentro de La Universidad de Cantabria. Por su labor, toda su entrega y las excelentes palabras que tiene siempre conmigo, gracias por apostar por mi.

Finalmente, tanto este trabajo, como todo en general, nada va a tener sentido sin el apoyo y el amor de mis padres, mis abuelos y de Sandra. Muchas gracias por estar incondicionalmente ahí.

## Resumen

Este trabajo fin de Máster se ha desarrollado a partir de la colaboración entre el Departamento de Ingeniería de Comunicaciones de la Universidad de Cantabria y el Consorcio ESS Bilbao que tuvo lugar de febrero a junio de 2019. Apoyado por este marco colaborativo, el estudio se centra en tres áreas clave: proporcionar un soporte técnico in situ para el proceso de reacondicionamiento de amplificadores klystron; contribuir a nuevas mejoras e iniciativas; para crear valor añadido y ganar experiencia en el campo científico y tecnológico de aceleradores de partículas.

Se han llevado a cabo diferentes desarrollos y mejoras en las instalaciones de prueba de radiofrecuencia del Consorcio ESS Bilbao, enmarcados en la optimización de los procesos de reacondicionamiento de amplificadores klystron de alta potencia. Se han llevado a cabo simulaciones circuitales y cambios de hardware en el sistema de protección de máquina. Se ha diseñado y desarrollado un nuevo software de medidas automáticas para la caracterización más ágil del comportamiento del klystron. También se ha elaborado un software para la medida de altos niveles de potencia de RF aplicando calorimetría. Estas actividades siempre se llevaron a cabo bajo las recomendaciones específicas establecidas de ESS y los requisitos técnicos.



# Abstract

This Masters dissertation has developed from the partnership between the Department of Engineering Communications at the University of Cantabria and ESS Bilbao Consortium which took place from February to June 2019. Supported by such collaborative framework, this study focuses on three key areas: to provide an in situ technical support for the Klystron amplifiers reconditioning process; to contribute to new improvements and initiatives; to create added value and gain experience in the scientific and technological field of particle accelerators.

Different developments and improvements have been carried out at the ESS Bilbao Consortium RF Test Stand, focused on the optimization of high-power klystron amplifiers reconditioning procedures. Circuit simulations and hardware changes in the machine protection system have been accomplished. New automated measurement software has been designed for an enhanced characterization of the klystron behaviour. In addition, a measurement software has been developed for an accurate measurement of high-power levels using calorimetry. Such activities were always performed under established ESS specific recommendations and technical requirements.

## Capítulo 1: Introducción

La Comisión Europea elaboró, en su hoja de ruta dentro del programa Horizonte 2020 sobre el aspecto de innovación, un plan basado en la construcción de una fuente europea de neutrones por espalación, basada en un acelerador lineal de partículas (o LINAC, del inglés *Linear Accelerator*), en el cual participasen diferentes países Europeos. Se propuso la candidatura española, a través del Consorcio ESS Bilbao para emplazar dicha instalación, liderada por el gobierno vasco, en conjunto con el Gobierno Central de España, a través del Ministerio de Ciencia, Innovación y Universidades. ESS Bilbao redefinió sus objetivos, tras decidirse que la fuente se construiría en Lund (Suecia), para llevar a cabo la contribución en especie a la fase de construcción de la Fuente Europea de Neutrones por Espalación. *European Spallation Source*, ó ESS ERIC (*European Research Infrastructure Consortium*).

La Fuente Europea de Neutrones por Espalación es un Consorcio de Infraestructura de Investigación Europea. Su objetivo principal es diseñar, planificar, construir, poner en marcha, operar y desmantelar la fuente europea de espalación (ESS), un centro de investigación a gran escala para la investigación de materiales.

ESS acogerá la fuente de neutrones más potente del mundo y aspira a ser uno de los centros líderes en el mundo para la investigación con neutrones en los campos de la química, la nanociencia, la energía, la ingeniería del medio ambiente, la alimentación, la biociencia, la farmacéutica, la informática y la ciencia e ingeniería de materiales.

ESS se sitúa en Lund, Suecia, coorganizada por Suecia y Dinamarca, y actualmente cuenta con 15 países asociados. ESS ERIC y sus socios están actualmente involucrados en la construcción de ESS, estando planeada su finalización a mediados de 2025. La instalación entregará sus primeros neutrones en 2023 y se espera que esté a pleno funcionamiento con 16 instrumentos para el año 2025.

El Consorcio ESS Bilbao fue creado en 2010 y está financiado por el Ministerio de Ciencia de España y la Comunidad Autónoma del País Vasco. Es la institución encargada

para realizar la contribución española en especie a la Fuente Europea de Neutrones por Espalación, ESS.

Además ESS Bilbao está capacitado para desarrollar proyectos de distinta índole dentro del campo de las tecnologías neutrónicas, y ofrecer así valor añadido en los diversos proyectos relacionados con tecnologías de partículas, contribuyendo positivamente en el desarrollo científico y tecnológico de nuestro país. Sus recursos humanos y tecnológicos han crecido sustancialmente con el propósito de consolidarse en el aporte científico y de investigación hacia la búsqueda constante de conocimiento en nuestra sociedad.

ESS Bilbao está a cargo de varios paquetes de trabajo para las principales divisiones científico-técnicas de ESS ERIC (Acelerador, Blanco, Controles e Instrumentos Neutrónicos Científicos). Dichos paquetes de trabajo son generalmente sistemas o subsistemas completos de gran complejidad técnica y alto valor añadido, abarcando desde el diseño, la construcción, la validación en las instalaciones del Consorcio ESS Bilbao, y la instalación y puesta en marcha en Lund (Suecia), e involucrando no sólo a los recursos de ESS Bilbao sino también a los agentes del sistema de ciencia vasco y español y a la industria de la ciencia.

Entre estos paquetes de trabajo se encuentran las estaciones de potencia de radiofrecuencia para el acelerador no superconductor. ESS utilizará un acelerador lineal (LINAC, del inglés *Linear Accelerator*) superconductor de pulso ancho y acelerará un haz de protones de 62.5 mA de intensidad hasta una energía de 1300 MeV. La potencia pico de haz que se enviará al blanco será de 125 MW, siete veces más que la infraestructura actual con potencia más alta.

En lo que respecta a los amplificadores de potencia, el Consorcio ESS Bilbao es responsable de suministrar un total de 6 klystrons de alta potencia. Tres de ellos son de nueva adquisición, mientras que los otros tres son procedentes de aquéllos adquiridos en 2010 a la empresa CPI (Communications & Power Industries, Inc).

Los klystrons adquiridos a CPI, referencia VKP-8352A, son básicamente similares a los necesarios para ESS (operación pulsada con potencia de pico de 2,9 MW a 352,21 MHz, ciclo de trabajo de 4,8%, niveles de alta tensión y corriente, eficiencia, equipos auxiliares), con la notable excepción de la duración de pulso y la frecuencia de repetición

de pulso. Los klystrons VKP-8352A se diseñaron para una operación a 1,6 mseg y 30 Hz, mientras que la fuente europea de neutrones requiere operación a 3,5 mseg y 14 Hz.

Comoquiera que ambas combinaciones de ancho de pulso y frecuencia de repetición arrojan un ciclo de trabajo similar (4,8-4,9%), el Consorcio ESS Bilbao consultó al fabricante sobre la viabilidad de una operación fiable en condiciones de pulso largo (3,5 mseg), a la vista de la previsible mayor variación de temperatura que sufrirían principalmente las piezas del colector.

De acuerdo con los análisis realizados por CPI, es posible operar en tales condiciones, siempre que previamente se haya comprobado el cumplimiento de las especificaciones originales, y asimismo se haya realizado un procedimiento de reacondicionamiento del klystron.

Dicho reacondicionamiento consiste básicamente en alimentar el klystron con pulsos de alta tensión de duración progresivamente creciente, observando los niveles de vacío (a fin de garantizar en cualquier caso un alto vacío, o bien cortar o disminuir la alta tensión en caso de detectarse picos debidos a liberación de tensiones o microfracturas que dieran lugar a desgasificación).

En este trabajo, en primer lugar, se presenta lo que es un acelerador lineal de partículas, sus partes fundamentales y se mencionarán ejemplos y aplicaciones en los distintos campos de la ciencia, para facilitar la comprensión y vision global de esta tecnología. En segundo lugar, se presentan más en detalle los sistemas vinculados a la radiofrecuencia (modulador, generadores de señal, amplificador driver de estado sólido, amplificadores klystron, guías de onda, detectores de potencia) como la generación y medición de las señales que intervienen en el subsistema completo de RF. En tercer lugar, se describe el desarrollo e implementación de las herramientas necesarias para la adquisición de medidas de radiofrecuencia de alta potencia, y con ellas facilitar al usuario el análisis.

La estructura por capítulos de este documento se compone de un **primer capítulo**, el presente, en el cual se contextualiza el trabajo desarrollado, describiendo el Consorcio ESS Bilbao, la institución ESS ERIC y se introduce el concepto de fuente de neutrones por espalación basada en acelerador lineal. Además, se enumeran los elementos básicos de un acelerador lineal y se describen algunos ejemplos de aceleradores en la industria.

En el **segundo capítulo** se describe un sistema acelerador más en detalle, cómo se genera y cómo se conduce la señal de radiofrecuencia, así como el componente en el que más nos centraremos a lo largo del trabajo, el amplificador de alta potencia klystron.

El **tercer capítulo** recoge un componente concreto que forma parte del puesto de test RF, el sistema de enclavamientos rápidos (fast interlocks), y las modificaciones en el hardware del mismo llevadas a cabo para su puesta a punto. En concreto, se describen los cambios realizados para el manejo de señales de muestra negativas de la tensión y la corriente del modulador, así como en los comparadores de potencia ubicados en las guías de onda de salida del klystron, para la corrección de una falsa alarma de potencia reflejada.

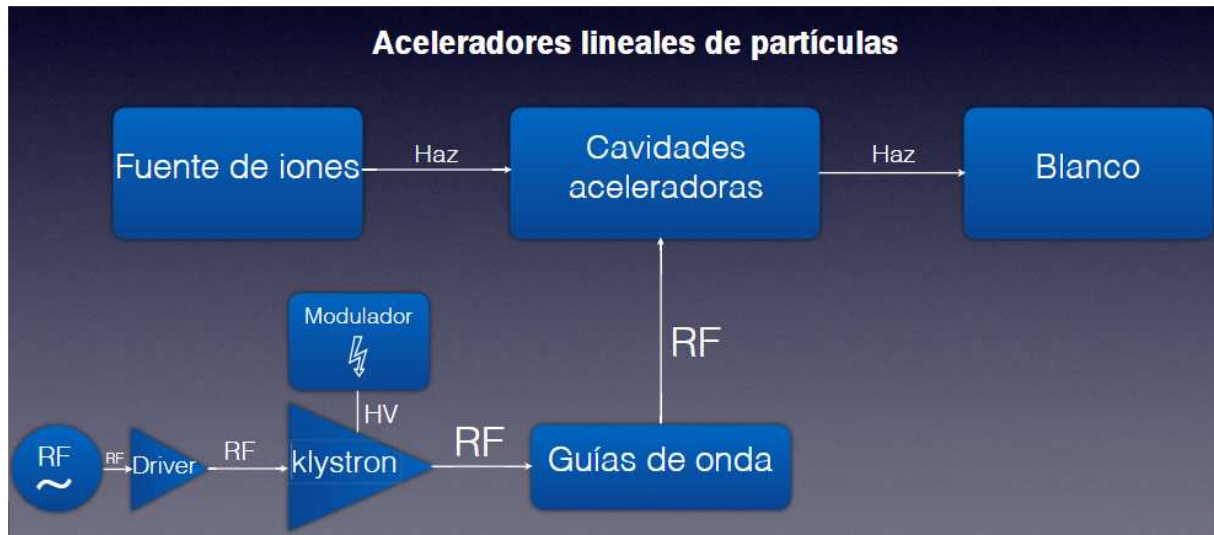
El **cuarto capítulo** se centra en el desarrollo de una interfaz gráfica de usuario en MATLAB, para las campañas de medidas de alta potencia de radiofrecuencia, permitiendo la caracterización en amplitud y en frecuencia.

El **quinto capítulo** describe el desarrollo de un software en LabView para la obtención de una medida más fidedigna de alta potencia de radiofrecuencia, mediante un método empleando calorimetría.

Finalmente en el **sexto y último capítulo**, se hace una valoración y se concluyen los aspectos fundamentales, así como una introducción a las posibles líneas futuras de investigación sobre los procedimientos que se desarrollaron en este trabajo.

## **1.1 Elementos básicos de un acelerador lineal de partículas**

Para describir las etapas de las que se compone un acelerador lineal de partículas se elabora un esquema clarificativo, exponiendo la función de cada elemento del acelerador. Estos sistemas se caracterizan por la función de cada una de sus secciones, la primera es la sección generadora (compuesta por la fuente de iones), después la sección aceleradora, constituida por las cavidades resonantes aceleradoras, y finalmente la sección del blanco o target donde colisiona el haz de iones. El subsistema de radiofrecuencia, compuesto por generadores de RF, amplificadores de potencia, moduladores, guías de onda), alimenta a las cavidades del acelerador. A continuación se representa el esquema de los componentes básicos de un acelerador lineal de partículas y una explicación resumida de los mismos.



**Figura 1.1** Esquema de un LINAC simplificado

- La *fente de iones* genera el haz de partículas. En muchos casos, en la fuente de iones se genera plasma en una cavidad sometida a vacío donde se inyecta hidrógeno. En condiciones de resonancia ciclotrón del electrón (ECR, *Electron Cyclotron Resonance*), determinada por el campo magnético y la potencia de radiofrecuencia, se genera plasma de alta densidad. Y después por diferencia de potencial electrostática se extraen los iones (protones).
- Una *cavidad* aceleradora de radiofrecuencia [14] es una cámara de metal, en cuyo interior existe alto vacío, y se halla un campo electromagnético oscilante a una frecuencia de resonancia específica. La aceleración de los iones es provocada por dicha oscilación del campo eléctrico.
- El *modulador* es una parte importante en el LINAC, ya que permite rectificar la corriente alterna de la red eléctrica de potencia, produciendo pulsos de alta tensión, para alimentar los amplificadores klystron. Algunos parámetros típicos de un modulador se presentan en la tabla 1.1:

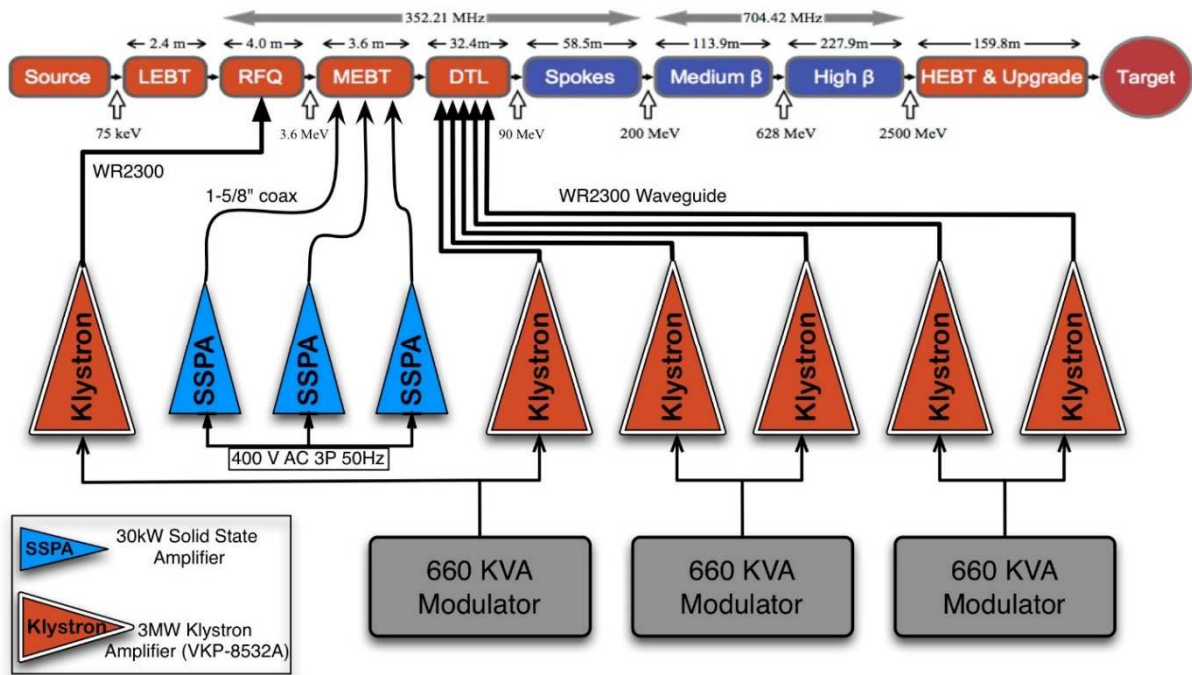
**Tabla 1.1** Parámetros del modulador de ESS Bilbao

Parámetro	Especificación
Voltaje de salida	120 kV (pulsado)
Corriente de salida	60 A
Anchura del pulso	1.81 ms (ajustables)
Frecuencia de repetición de pulso	2 - 50 Hz
Potencia pico	7.2 MW
Potencia promedio	648 kW
Ciclo de trabajo (%)	9%
Voltaje de entrada	30 kV (50 Hz)
Eficiencia	>90%

- El generador de RF proporciona una señal de un tono de alta frecuencia, la que será encargada de alimentar varios componentes del acelerador, para aumentar la velocidad de las partículas, una vez sea amplificada por el preamplificador driver y después por el klystron.
- El preamplificador (driver amplifier) [1], es un amplificador de RF de estado sólido encargado de aumentar la amplitud de la señal de radiofrecuencia hasta el valor adecuado a la entrada del klystron. En el caso de ESS Bilbao, el valor a la salida de esta etapa de amplificación es de aproximadamente 300 W de potencia pico.
- El amplificador de potencia es un klystron, es un tubo electrónico de vacío cuyo funcionamiento se basa en la modulación de la velocidad del haz de electrones.
- Las guías de onda permiten conducir la señal de radiofrecuencia amplificada por el klystron, y posteriormente inyectarla en las cavidades a través de un acoplador de potencia, introduciendo pocas pérdidas de inserción en el conjunto.
- El blanco o target, es el último elemento del acelerador a donde va dirigido el haz de partículas. En el caso de ESS, en él se producen los neutrones, por el fenómeno de espalación, es decir, por la colisión de los protones de alta velocidad sobre los núcleos de los átomos de Tungsteno.

## 1.2 Esquema del acelerador de protones de la fuente ESS

El acelerador lineal de protones de ESS [1] presentará una longitud de 602.5m de largo, cuyas secciones se describen brevemente en el esquema de la Figura 1.2.



**Figura 1.2** Secciones del acelerador de protones de ESS y estaciones de RF para la sección no superconductora.

En la Figura 1.2, las partes de color naranja de la línea de acelerador (Source, LEBT, RFQ, MEBT, DTL, HEBT) indican las secciones conductoras a temperatura ambiente. Por otro lado, las partes de color azul oscuro (Spokes, Medium  $\beta$  y High  $\beta$ ) representan secciones superconductoras.

A continuación, se describen las diferentes secciones del acelerador de protones de ESS:

- La fuente de iones es una cámara de descarga, en cuyo interior existe un campo electromagnético que oscila a frecuencia de microondas, el cual en presencia de moléculas de  $H_2$  genera un plasma de protones.



- La línea de transporte de haz de baja energía LEBT (Low Energy Beam Transport) es la sección de diagnóstico de haz que conduce al primer elemento acelerador, el cuadrupolo de radiofrecuencia o RFQ (Radio Frequency Quadrupole).
- El RFQ tiene tres funciones fundamentales: i) acelerar el haz extraído del inyector (de 70 keV a 3.62 MeV). ii) Enfocar el haz (sin necesidad de imanes). iii) Agrupar o empaquetar las partículas (formar “bunches”).
- En la línea de transporte de haz de media energía o MEBT (Medium Energy Beam Transport) se cumplen cuatro funciones diferentes, posee óptica para adaptar y dirigir el haz desde el RFQ hacia el interior del DTL (Drift Tube Linac), tiene un conjunto de dispositivos de instrumentación de haz, permite colimación de la distribución transversal de las partículas. El chopper<sup>1</sup> del MEBT es más rápido que el del LEBT. El DTL es el encargado de elevar la energía del haz de protones de 3.62 MeV a 90 MeV. La primera sección de superconducción está constituida por cavidades de tipo doble spoke con un valor  $\beta$  geométrico de 0.50.
- Las secciones superconductoras Medium $\beta$  y High $\beta$  están formadas por cavidades elípticas que aceleran las partículas hasta 1.3 GeV.
- El HEBT (High Electron Beam Transport) es la última etapa que conduce el haz de alta energía hasta el blanco de espalación rotatorio ubicado en el final del recorrido del haz.
- El blanco o *target* es el último elemento del acelerador, formado por una estructura giratoria en forma de rueda, fabricada a partir de tungsteno.

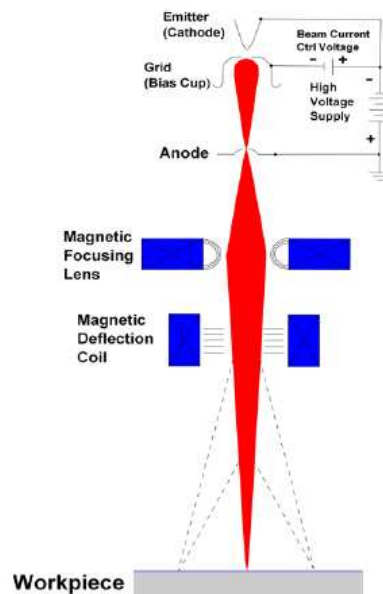
---

<sup>1</sup> Chopper: es un mecanismo de corte o interrupción del haz de electrones que se encuentra dentro del MEBT y del LEBT para cambiar la intensidad del haz o para evitar pérdidas críticas y evitar la colisión con la estructura del target rotatorio cuando sea necesario.

<sup>2</sup>  $\beta$  : es la fracción de la velocidad de la partícula respecto a la velocidad de la luz  $\beta = \frac{v}{c}$ , siempre adquiere un valor entre 0 y 1 [7] .

### 1.3 Ejemplos de aplicaciones industriales y científicas de los aceleradores de partículas

Dentro de las aplicaciones que se pueden encontrar tanto en el campo de la industria, como en los ámbitos de la medicina y de la investigación, se pueden recoger las siguientes utilidades de los aceleradores de partículas. Una de ellas es el empleo de haces de electrones para llevar a cabo tareas de procesamiento de materiales. Inicialmente [2] fue un proceso patentado por Pirani en 1907, que consistía en utilizar un haz de electrones como fuente de calor para la fusión de metales refractarios en vacío. El proceso empleaba dos tecnologías fundamentales, la ingeniería del vacío y la óptica electrónica. Ambas alcanzaron una gran madurez proporcionando las bases para fabricar sistemas de procesamiento de materiales basados en haces de electrones. Un esquema ejemplo del sistema de procesamiento de materiales utilizados en la actualidad, se representa en la siguiente Figura 1.3.



**Figura 1.3** Esquema que muestra los componentes básicos y el funcionamiento de los sistemas de procesamiento de materiales por haz de electrones [2].

La soldadura por haz de electrones es una tecnología que se adquirió con objeto de investigación, pero más tarde, ha sido potenciada para aplicarla en proceso de manufacturación aeronáutica y toda la industria aeroespacial en general.

En la imagen anterior observamos las partes del soldador basado en acelerador de partículas, la parte de inyector (*emitter*), posteriormente los elementos que crean el campo magnético de contención (*magnetic focusing lens*), y la bobina deflectora (*magnetic deflection coil*).

En el ámbito relacionado a la fabricación de semiconductores [2] se establecen una serie de condiciones para acelerar iones para dopar y realizar modificaciones en ciertos materiales basados en semiconductores. Esta técnica se conoce como implantación de iones, se puede considerar como una técnica básica para la fabricación de circuitos integrados, esenciales hoy en día para el desarrollo y la innovación en áreas tan diversas como las comunicaciones o la medicina.

Además, existe otra técnica basada en la irradiación de materiales y productos comerciales con energía ionizante en forma de electrones acelerados y rayos X, los cuales emiten los electrones energéticos, además de emisión de rayos gamma, que son emitidos por los nucleidos. Dicho método es conocido por procesamiento de radiación. Se puede modificar por este método las características físicas, químicas, o biológicas de los materiales.

En medicina, son comunes los isótopos radiactivos, radionucleidos o radioisótopos. Las aplicaciones de los isótopos como radiotrazadores, cuyo primer científico que la aplicó en 1911 fue George de Hevesy (considerado el padre de la medicina nuclear).

Las aplicaciones más conocidas en este ámbito son la imagen para el diagnóstico y la radioterapia para tratamientos oncológicos. Los dispositivos aceleradores de partículas permiten emitir tratamiento de radiación con tecnología más apropiada para la generación de imágenes [4], dirigido localmente a la zona de interés del cuerpo del paciente.



**Figura 1.4** Acelerador lineal para tratamiento médico

En 1919, Ernest Rutherford, (galardonado con un premio Nobel en 1908 por su investigación en radioactividad), fue el primer científico que reconoció que un elemento se podía transformar en otro a través de medios artificiales. Puso en evidencia que bombardeando a gas de nitrógeno con partículas alfa, desde una fuente radioactiva, se observaba que una partícula alfa se detenía ocasionalmente, y al mismo tiempo producía la emisión de un protón con elevada energía cinética. Este experimento, dio lugar a la primera demostración de reacción nuclear artificial, es decir, no provocada por causas naturales. La investigación de Rutherford se veía limitada debido a emplear un número bajo de partículas alfa de poca energía procedentes de fuentes radiactivas. En 1927, Rutherford requirió un dispositivo que fuese capaz de producir partículas suficientemente energéticas, y expuso dicho reto a todo tipo de investigadores para realizar un sistema generador. Esta propuesta permitió que se desarrollaran unos sistemas electrostáticos de alta tensión, creados por Cockcroft-Walton y Van de Graaff, pero se encontraban restringidos por la energía o corriente de haz que eran capaces de generar. Fue a partir de entonces, cuando los conceptos de aceleradores lineales se comenzaron a concebir. No obstante, se consideraban poco prácticos de generar o necesitados de fuentes de RF que por aquel entonces, no estaban disponibles. El desarrollo revolucionario llegó en 1929-

1930 mediante la invención del ciclotrón<sup>3</sup> por Ernest O. Lawrence, a partir de las ideas de Rolf Wideroe.

Actualmente, a pesar de que los ciclotrones, son los sistemas predominantes para la producción de radionucleidos, los aceleradores lineales de RF han sido también desarrollados para este propósito. Estos aceleradores lineales están limitados a un solo haz, pero el reducido blindaje requerido y la estabilidad de operación hacen que presenten ventajas sobre los ciclotrones.

---

<sup>3</sup> Ciclotrón: es un tipo de acelerador de partículas que a diferencia de los aceleradores lineales, consta de dos placas huecas con forma de semicircunferencia o también conocidas como ‘Des’, donde en su interior se aplican oscilaciones de alta frecuencia, se aceleran las partículas con trayectoria circular. Posteriormente son dirigidas a la zona intermedia de separación [3].

## Capítulo 2: Sistemas de radiofrecuencia para aceleradores

La fuente de neutrones de ESS ERIC [1] está basada en un acelerador lineal de protones de alta energía y operación pulsada que incluye una sección normalmente conductora y otra superconductora.

La sección normalmente conductora está formada por cavidades resonantes fabricadas en cobre (por su alta conductividad) que operan a temperatura ambiente. Se trata de un RFQ (Radio Frequency Quadrupole, Cuadrupolo de Radio Frecuencia), un MEBT (Medium Energy Beam Transport, Línea de Transporte de Haz de Media Energía) y un DTL (Drift Tube Linac, Acelerador de Tubos de Deriva) que aceleran las partículas producidas extraídas de la fuente de iones hasta una energía de 90 MeV.

Al tratarse de cavidades no superconductoras, una parte importante de la potencia de RF se disipa en las paredes de las propias cavidades y no es útil para acelerar las partículas. Como resultado, habitualmente se requieren potencias de RF pulsada de incluso megavatios de pico para alimentar las cavidades. Los sistemas transmisores de RF de alta potencia que alimentan las cavidades de los aceleradores de partículas son críticos tanto por su complejidad técnica, por sus elevados costes de adquisición y operación, y por su fiabilidad. Es necesario realizar un buen diseño considerando cuidadosamente el balance de costes, prestaciones técnicas, eficiencia energética, fiabilidad, disponibilidad, estrategias de mantenimiento e inspección, etc.

El Consorcio ESS Bilbao es responsable, como parte de la contribución española a la Fuente Europea de Neutrones ESS ERIC, del desarrollo de las estaciones de RF para el acelerador a temperatura ambiente (RFQ, MEBT y DTL), así como el LLRF de la sección superconductora spoke y las cavidades buncher, operando todo ello a 352 MHz.

Dados los requisitos de potencia de RF (cerca de 3 MW de pico, en las cavidades tipo RFQ y DTL) a frecuencias relativamente bajas (352 MHz), prácticamente la única opción para los transmisores es el empleo de klystrons de potencia. Estos dispositivos son tubos electrónicos de vacío capaces de proporcionar muy altas potencias de RF a partir de la modulación de velocidad de un haz de electrones en su interior.

Para la operación del klystron es necesario alimentarlo con alta tensión pulsada (del orden de 110 kV) y alta corriente (cercana a 50 A). Las fuentes de alta tensión pulsada son conocidas como moduladores, e incluyen etapas de electrónica de potencia de baja tensión y de alta tensión (rectificadores, convertidores DC/DC, bancos de condensadores, inversores, transformadores de alta tensión, rectificadores de salida, filtros), todas ellas perfectamente sincronizadas para asegurar una alta calidad del pulso de salida.

Para una adecuada aceleración del haz de protones, es necesario asegurar unas características de estabilidad de amplitud y fase de la señal de RF, lo cual se consigue con un sistema de control digital en lazo cerrado llamado LLRF (*Low Level RF Control*). Este sistema se encarga de tomar una muestra del campo acelerante en la cavidad y compensa las desviaciones puntuales de amplitud y fase modificando la señal aplicada al transmisor de potencia. Asimismo, se encarga del control de la frecuencia de resonancia de la cavidad, bien por medio de sintonizadores mecánicos, o por los sintonizadores piezoeléctricos (como es el caso de las cavidades superconductoras).

Análogamente, las cavidades de tipo buncher (cuyo objetivo es “empaquetar” los iones), requieren potencias de RF que rondan los 30 kW de pico. En este caso, los amplificadores de estado sólido (SSPA, *Solid State Power Amplifier*) basados en transistores LDMOS empiezan a constituir una excelente alternativa a los tubos electrónicos de vacío, compitiendo en términos de coste, eficiencia, fiabilidad... Los SSPA se alimentan directamente desde corriente alterna, y no requieren ninguna precaución de vacío o de alta tensión como los tubos. Los niveles de potencia citados permiten que la distribución se realice mediante líneas coaxiales rígidas de dimensiones más manejables que las guías de onda rectangulares.

El desarrollo de cavidades resonantes aceleradoras incluye la realización de diseños electromagnético, térmico, mecánico, de vacío, así como la fabricación y las pruebas de caracterización y validación de RF en baja y en alta potencia (modos resonantes, factor de calidad, factor de acoplo, mapa de campos con bead pull,...).

En la Figura 2.1 se pueden apreciar las partes fundamentales de un sistema de RF, para un acelerador de partículas. En la imagen, se exponen los elementos que se emplean para realizar una contribución muy importante de ESS Bilbao a la fuente ESS ERIC, con el reacondicionamiento de los amplificadores klystron a operación con pulso largo. Se

muestra la cadena de alta potencia, que se representa mediante líneas sólidas, y la parte de baja potencia con líneas discontinuas. Para acelerar los iones, se requieren diferentes tipos de estructuras aceleradoras, porque cada tipo de estructura tiene una aceptación limitada de velocidad de la partícula.

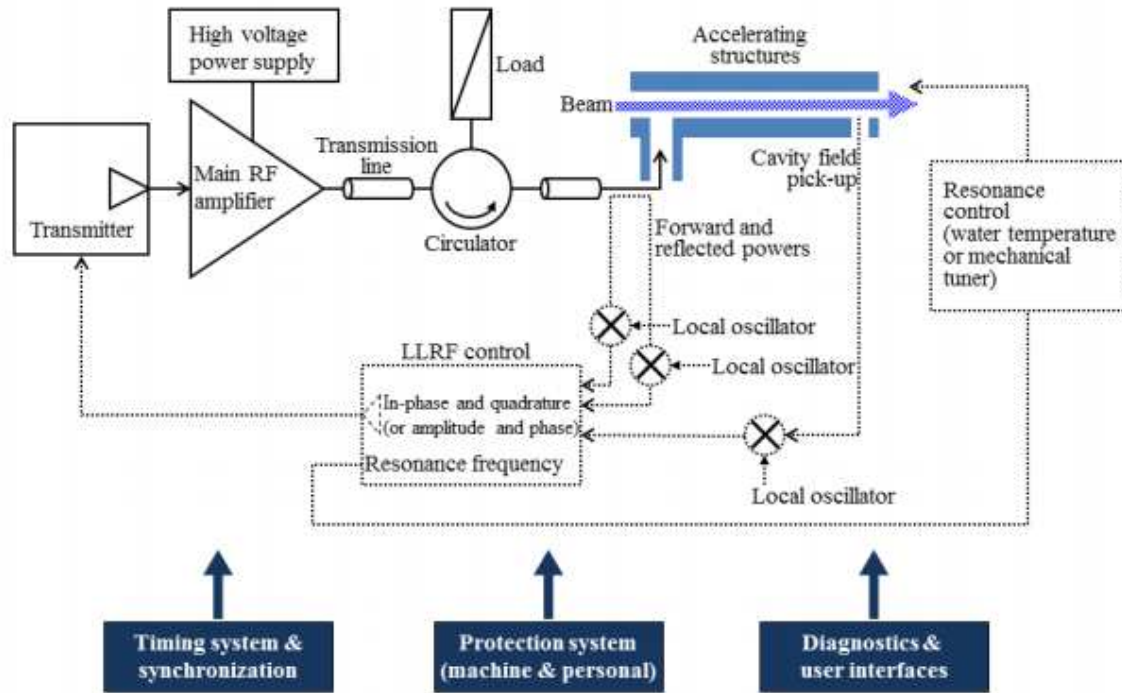


Figura 2.1 Sistema de radiofrecuencia típico para acelerador lineal básico [16]



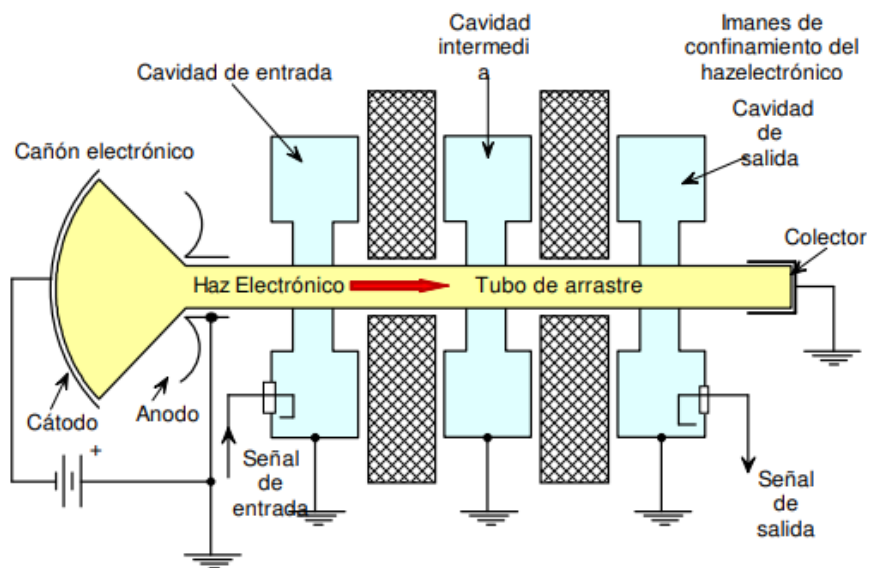
## 2.1 Klystron

En este apartado se describe el elemento que amplifica la radiofrecuencia antes de introducirse en las cavidades. El dispositivo conocido como amplificador klystron.

Un klystron se define como tubo o válvula de vacío [17] , que se emplea para generar y aumentar la amplitud de señales de altas frecuencias. Fue diseñado por primera vez por los hermanos Russell y Sigurd Varian en el año 1937. Puede operar de dos maneras, como elemento amplificador, o como un oscilador. El principio de funcionamiento del klystron es la modulación de la velocidad de un haz de electrones, los cuales aceleran y desaceleran en función de una señal oscilante en el tiempo.

Respecto a la función del klystron como amplificador, en su versión más simple, consiste en un tubo electrónico con varias cavidades. En la Figura 2.2, se muestra un klystron con tres cavidades. La primera cavidad, es la de entrada, a ésta se le aplica una señal de radiofrecuencia. A continuación, una cavidad intermedia (puede haber más cavidades intermedias), y finalmente la cavidad de salida de la señal amplificada. Todas esas cavidades van conectadas por tramos conocidos como tubos de arrastre. Para que no se propague señal electromagnética por los tubos, y por ello se pierda el aislamiento entre las cavidades, se diseñan dichas cavidades de manera que no propaguen señal a la frecuencia de funcionamiento de los tubos. Esto es fundamental en un amplificador de potencia.

El haz de electrones se genera en el cañón electrónico basado en un filamento de tungsteno caliente. Dicho haz se acelera mediante una elevada tensión aplicada en el ánodo y posteriormente transcurre a través de los tubos de arrastre, frente a las cavidades hasta que impacta en el colector. El cuerpo principal del tubo, incluyéndose el colector, están conectados a tierra, mientras que el cátodo y los electrodos que enfocan el haz se someten a una alta tensión negativa entre -60 y -120 kV (en su uso para aceleradores de partículas).



**Figura 2.2** Estructura básica de un klystron de tres cavidades [17]

Se observa tanto en la Figura 2.3 como en la Figura 2.4, uno de los klystron fabricados por la empresa CPI (Communications & Power Industries) en las instalaciones del Consorcio ESS Bilbao, donde se les realizaron los procedimientos de reacondicionamiento para su posterior envío a la sede de ESS en Lund. El klystron que se reacondiciona en las instalaciones de ESS Bilbao, presenta las partes propias de estos sistemas, el cañón de electrones, las cavidades resonantes, y el colector. Dispone también de un solenoide, formado por electroimanes que se encargan de mantener el haz de electrones confinado a lo largo del tubo de arrastre.



**Figura 2.3** Klystron de CPI, ESS Bilbao



**Figura 2.4** Klystron con el blindaje incorporado, ESS Bilbao.

Estos dispositivos no sólo se utilizan en aceleradores de partículas sino que también se emplean en aplicaciones de comunicaciones satelitales o de radiodifusión ya que proporcionan gran potencia de salida en las bandas de radiofrecuencia. Por contra, generan un coste añadido al funcionamiento del sistema, debido al alto consumo. Los klystron que analizaremos operan en amplificación de clase A. El modo de operación en la instalación es de onda pulsada según requerimientos de ESS ERIC. Es destacable que en términos de disipación de potencia, se consiguen unos valores que alcanzan entorno al 55% de eficiencia [1].

Para proporcionar según lo establecido, una potencia de pico de  $RF = 2,9 \text{ MW}$ , ancho de pulso de  $RF$  de 3,5 ms y 14 Hz de frecuencia de repetición de pulso de  $RF$ , es necesario generar una tensión de unos 100 kV de amplitud y una corriente de casi 50 A con el modulador. Se observa en la siguiente Figura 2.5 la entrada de alta tensión negativa en el cátodo del klystron.

El cañón de electrones del klystron de ESS Bilbao está basado en un filamento de tungsteno que es calentado por una fuente de corriente de unos 25 A.

El solenoide, por otro lado, tiene tres fuentes de corriente para alimentar los electroimanes. La primera se encarga de focalizar el haz y las otras dos de mantener el

haz confinado a lo largo del tubo de arrastre. Se emplean tres fuentes idénticas que operan a unos 10-12 A, siendo el valor correcto de cada una fijado por el fabricante del klystron.

Las dos bombas iónicas de vacío se encargan de mantener un alto nivel de vacío en el interior del klystron (mejor que  $10^{-8}$  mbar). Están alimentadas por sendos controladores que proporcionan una tensión de unos 5 kV y una corriente que es dependiente del nivel de vacío, lo cual permite monitorizar la presión a través de la corriente medida. Corrientes altas se corresponden con presiones altas, es decir, bajo vacío. Corrientes bajas se corresponden con presiones bajas, esto es, alto vacío.



**Figura 2.5** Entrada de alta tensión negativa en el klystron

## 2.2 RF test stand

La instalación de pruebas de RF de alta potencia de ESS Bilbao en Zamudio (RF Test Stand) consiste en un recinto cerrado de 160 metros cuadrados con sistema de protección de personas, fuente de alta tensión, klystron, circulador, carga, guías de onda, LLRF e interlocks (enclavamientos del sistema de protección de máquina), refrigeración líquida de alta pureza, etc. Permite realizar pruebas y acondicionamiento de componentes de RF en alta potencia (como el reacondicionamiento de klystrons para operación en pulso largo, cavidades buncher, acopladores de potencia, circuladores, entre otros).

El RF Test Stand está rodeado de un área de seguridad para evitar que las personas estén cerca de la instalación de pruebas mientras está en funcionamiento. Es importante delimitar la zona debida al riesgo que existe por la posible emisión de rayos X o Gamma, en algunas ocasiones. Sin embargo, los valores observados de radiación son mínimos. Por ello, se establece un vallado metálico, con cierre de seguridad. Esta zona se denomina (*human safety area*).

Para una correcta interpretación del funcionamiento de la instalación, por parte de los que estén trabajando en ese lugar, se encenderán unas luces fácilmente identificables antes de la operación de los instrumentos, que indicarán si está siendo utilizado o no. Se facilitan tres pulsadores por el área de operación, donde la persona encargada de revisar que no queden personas en las zonas circundantes debe ir pulsando secuencialmente. Entonces, el operario cierra la puerta del área de seguridad hasta que acabe de trabajar con los sistemas. Existen además botones de parada de emergencia, muy comunes en este tipo de instalaciones.

A continuación se describen brevemente los elementos que forman parte del RF test stand:

- Fuente de alta tensión pulsada o modulador, que proporciona hasta 120 kV, 60 A, 3,5 ms de duración del pulso, 14 Hz de frecuencia de repetición de pulso, a partir de la alimentación trifásica de 400 Vac.
- Klystron de potencia pulsado de 3 MW de pico, 150 kW de promedio, a 352 MHz.

- Equipos auxiliares para el klystron, incluyendo la fuente del filamento, las fuentes del solenoide, los controladores de las bombas iónicas de vacío y otros.
- Red de guías de onda rectangulares WR2300, circulador y cargas de RF de agua para 3 MW de pico, 150 kW promedio, a 352 MHz.
- Sistema de generación y distribución RF de baja potencia.
- Sistema de control de la instalación, con la máquina de estados, en entorno LabView.
- Sistema de refrigeración con circuitos de agua desionizada (para el klystron) y desmineralizada (para el modulador y las cargas), controlado por PLC.
- Sistema de protección de máquina (enclavamientos o interlocks), para la operación segura de los equipos de la instalación y apagado/corte rápido/lento en caso de fallos.
- Sistema de seguridad de personas, para garantizar la protección de los operadores de la instalación ante riesgos eléctricos, de radiación no ionizante y de radiación ionizante.
- Equipos de medida y prueba de RF (analizador de espectros, analizador de redes, analizador de espectros, osciloscopio, etc.)

La visualización e interacción con el usuario de todo el proceso se lleva a cabo a través de LabView. Ciertos aspectos fundamentales como son el consumo y de la potencia eficaz de RF que éste entrega a la siguiente sección son también monitorizados. Para ello, se utilizan detectores de potencia en todo el sistema de generación de onda pulsada, desde el amplificador de estado sólido (driver) hasta las guías de onda que transportan el haz hasta la siguiente etapa del acelerador. Por ello, mediante uso de las sondas de potencia y de diversos sensores distribuidos por todo el RF test stand, podemos estimar qué potencia incide y qué potencia se refleja en tiempo real. En línea con la protección frente



a potencia reflejada elevada, se fijan umbrales de corte de RF respondiendo a nuestras necesidades, a través del sistema de enclavamientos rápidos o Fast Interlocks.

La problemática reside en el proceso de calibración de los componentes, los cuales presentan pérdidas desde que la onda es medida hasta que es procesada por el software de medida, o por el osciloscopio. Esto traslada unas décimas de voltios que se traducen en decenas o centenas de vatios de potencia. Es imprescindible caracterizar todos los componentes, para cuantificar dichas pérdidas.

Esta instalación de pruebas de RF permite la realización de pruebas que abarcan por ejemplo la caracterización de componentes pasivos de RF en guía de onda WR2300 y línea rígida coaxial EIA 1-5/8 inch (por ejemplo: tramos rectos/curvos/codos, tramos flexibles, acopladores direccionales, desfasadores, conmutadores, divisores de potencia, transiciones, cargas, circuladores, etc.), así como la caracterización en baja y en alta potencia y el acondicionamiento de cavidades aceleradoras (buncher, RFQ). A continuación se muestra una imagen de la instalación RF test stand en ESS Bilbao:

Una vez realizados los procesos de reacondicionamiento en ESS Bilbao, los klystron fueron enviados a las instalaciones de ESS ERIC en Suecia.



**Figura 2.6** Test Stand de RF de alta potencia en Zamudio



**Figura 2.7** Klystron instalado en ESS ERIC (Lund, Suecia)



## Capítulo 3: Mejoras introducidas en los sistemas de enclavamientos rápidos

En este capítulo se documentan las modificaciones y las mejoras realizadas en los comparadores de umbral del sistema de enclavamientos rápidos (*fast interlocks*) del RFTS (Test Stand de RF de alta potencia) del Consorcio ESS Bilbao.

### 3.1 Sistema de enclavamientos rápidos de ESS Bilbao.

Antes de entrar en detalle en los cambios realizados en el sistema de enclavamientos rápidos (*fast interlocks*), realizaremos una breve introducción a los mismos. El sistema de enclavamientos [1] monitoriza y supervisa la prevención de daños a los equipos de radiofrecuencia dentro de la cadena de RF. Además, recibe por sus entradas analógicas y digitales las señales provenientes de los sensores del RF test stand que detectan los posibles fallos de los componentes. Las señales que se monitorizan se dividen en señales rápidas y señales lentas, dependiendo del tiempo de respuesta requerido. El sistema de enclavamientos rápidos de ESS Bilbao, consiste en un circuito electrónico basado en puertas lógicas digitales y comparadores de señales analógicas, realizados con amplificadores de instrumentación, diseñados según los criterios de tiempos de respuesta impuestos por ESS ERIC. En el caso del *fast interlocks* el tiempo de respuesta es inferior a 10 microsegundos.

El sistema de interlocks consta de las siguientes funcionalidades:

- Entradas analógicas, con comparadores de umbral o de ventana, optoacopladas.
- Entradas digitales optoacopladas.
- Un latch tipo D octal, para almacenar la condición de fallo.
- Una entrada de disparo (trigger), con monoestables para recuperar los flancos de subida y de bajada.
- Una entrada de reset.

Aludiendo a las condiciones de protección, se realizará una interrupción de la unidad de RF y/o del modulador de alta tensión dependiendo de la condición del fallo.

El sistema debe mantener un registro del último evento de enclavamiento relevante para permitir el análisis post mortem. Además, las funciones de enclavamiento deben realizarse independientemente de las funciones de control de RF y no deben verse afectadas por ninguna latencia en el tratamiento de las señales o el software.

El conjunto de señales que recoge el sistema *fast interlocks* [8] son las que se mencionan a continuación:

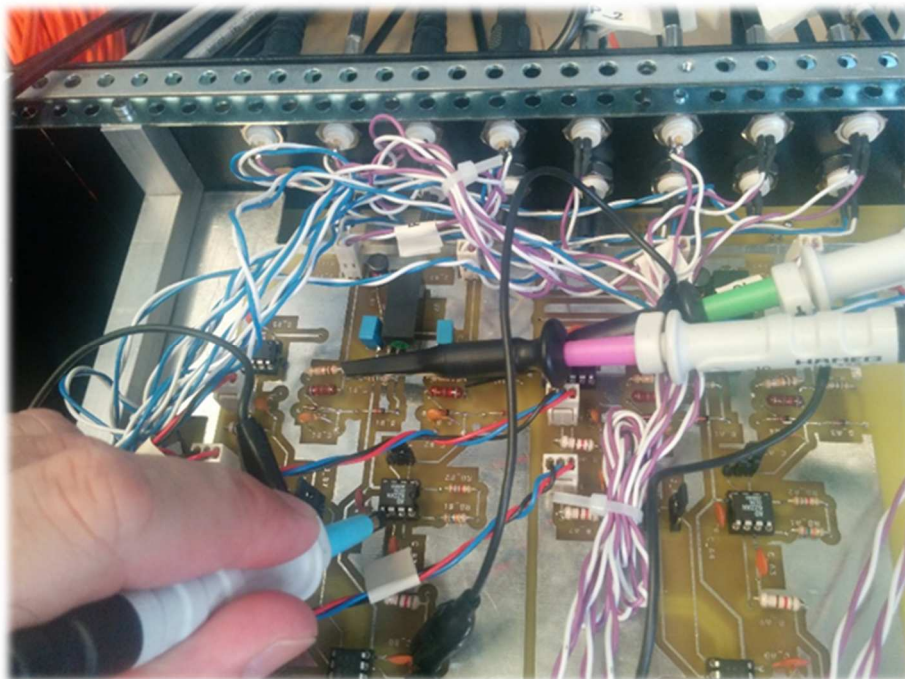
- *Beam Current* (corriente de haz): es una señal analógica que se encarga de controlar el valor de la corriente del haz de electrones del klystron, proveniente del modulador.
- *Beam Voltage* (tensión de haz): es una señal analógica que recoge el nivel de tensión del haz de electrones del klystron. También proviene del modulador.
- *Vacuum Ion pump #1* (bomba iónica de vacío #1): es una señal analógica que se recibe del controlador de la bomba iónica de vacío #1. Esta señal analógica es proporcional a la corriente que consume la bomba iónica, que se corresponde con el nivel de vacío. A mayor valor de corriente hay mayor presión, por lo tanto, un vacío más bajo.
- *Vacuum Ion pump #2* (bomba iónica de vacío #2): es una señal analógica que se recibe del controlador de la bomba iónica de vacío #2.
- *Solenoid #1* (Solenoides #1): es una señal analógica procedente de la fuente de corriente del solenoide #1.
- *Solenoid #2* (Solenoides #2): es una señal analógica que procede de la fuente de corriente del solenoide #2.
- *Gun PS* (Enfoque del cañón de electrones): es una señal analógica relativa a la fuente de corriente que enfoca el haz del cañón de electrones.
- *RF Overdrive* (SSPA) (exceso de RF en el preamplificador de estado sólido): es una señal analógica que da cuenta del posible exceso de potencia de RF en el preamplificador o driver.

- *RF Overdrive* (Klystron) (exceso de RF en el Klystron): es una señal analógica que da cuenta del posible exceso de potencia de RF en el Klystron.
- *Reflected Power* (SSPA) (Potencia reflejada en el preamplificador de estado sólido): es una señal analógica que es proporcional a la potencia de RF reflejada en el preamplificador o driver.
- *Reflected Power* (Klystron) (Potencia reflejada en el Klystron): es una señal analógica que es proporcional a la potencia de RF reflejada en el Klystron.
- *Arc detector* (Klystron) (Detector de arco eléctrico en el Klystron): es una señal digital que indica si hay arco eléctrico en la guía de onda de salida del Klystron.
- *Arc detector* (Circulator) (Detector de arco eléctrico en el circulador): es una señal digital que indica si hay arco eléctrico en el circulador.
- *Arc detector* (Coupler #1) (Detector de arco eléctrico en el acoplador #1): es una señal digital que indica si hay arco eléctrico en el acoplador #1.
- *Arc detector* (Coupler #2) (Detector de arco eléctrico en el acoplador #2): es una señal digital que indica si hay arco eléctrico en el acoplador #2.
- *Multipactor* #1: es una señal digital que da cuenta de si se produce efecto multipactor.
- *Multipactor* #2: es una señal digital que da cuenta de si se produce efecto multipactor.

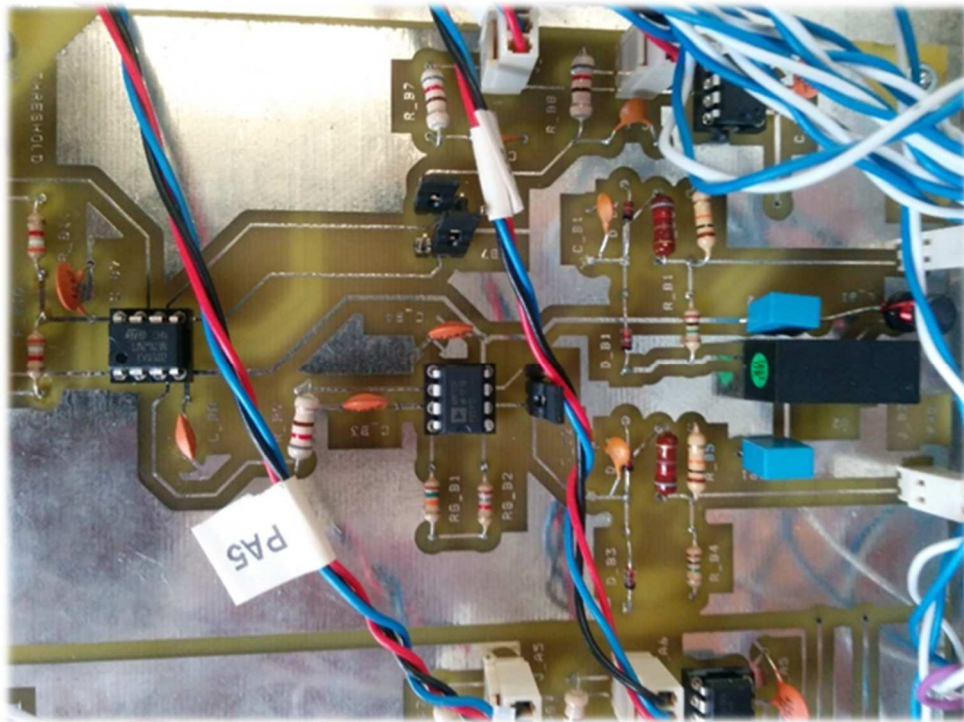
A continuación, se presentan las imágenes que muestran el aspecto del módulo del sistema fast interlocks, del exterior y de su circuitería interna.



**Figura 3.1** Frontal del sistema fast interlocks



**Figura 3.2** Circuito del comparador de umbral del sistema fast interlocks



**Figura 3.3** Circuito del comparador de umbral del sistema fast interlocks

En cuanto a los sistemas de interlocks de detectores de arco en las guías de onda, el Consorcio ESS Bilbao dispone de varias unidades basadas en detección de luz con fibra óptica, para detección de arco eléctrico en el interior de algunos componentes de RF en los que se pueda superar la tensión de ruptura, por ejemplo, en el interior de las guías de onda.



**Figura 3.4** Detectores de arco, controladores de las bombas iónicas de vacío y fuente de alimentación del filamento.



**Figura 3.5** Pruebas de validación en ESS Bilbao del sistema detector de arcos para el RFQ de ESS ERIC

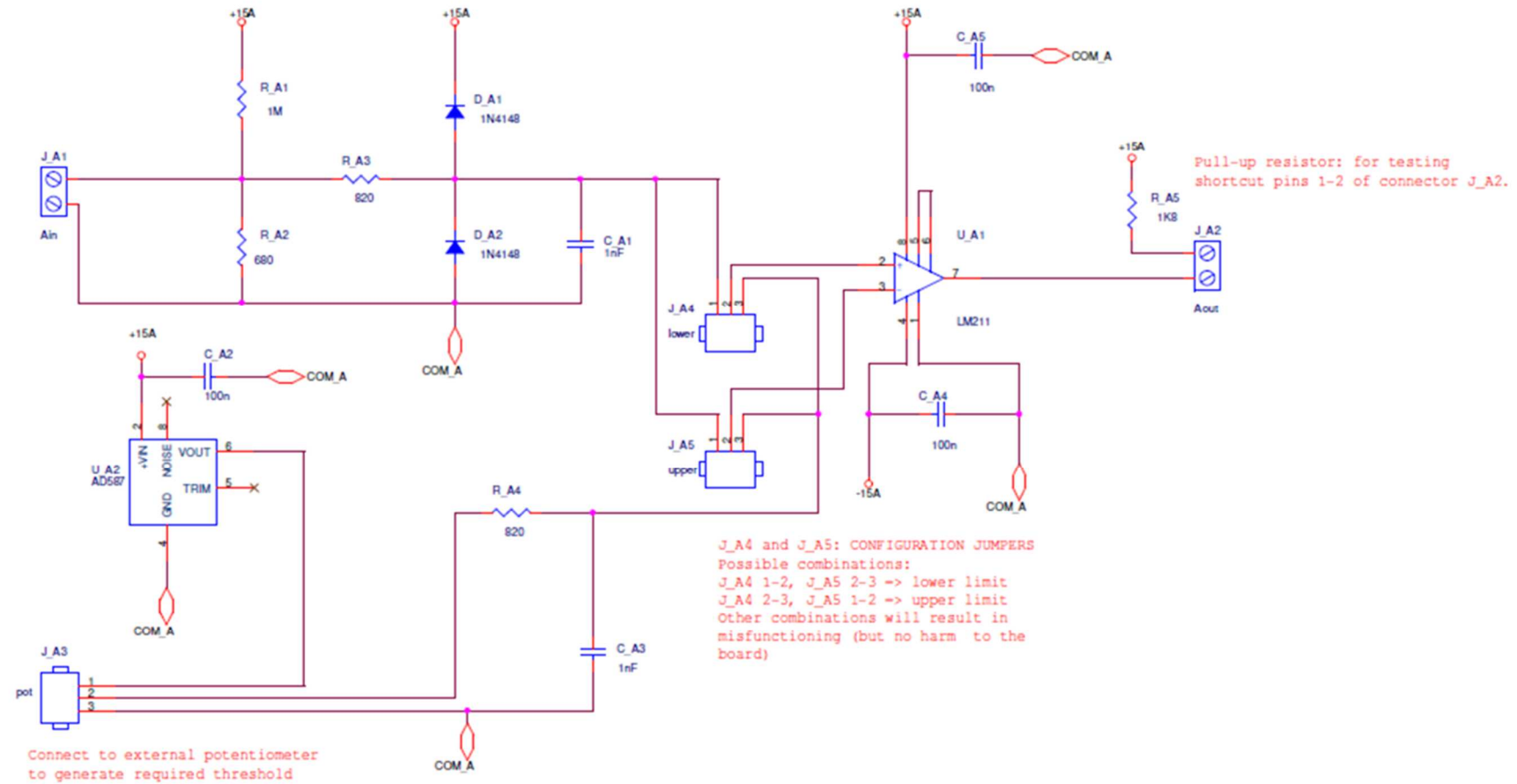


## 3.2 Comparadores de umbral del sistema fast interlocks

En este elemento tan importante de protección con respuesta rápida, se realizaron una serie de ajustes para que se pudieran llevar a cabo las medidas del klystron sin falsas indicaciones de alarma.

En la Figura 3.6 se detalla un esquemático del comparador de una entrada analógica (canal A). El funcionamiento es el siguiente, se filtra la señal de entrada por una red RC ( $R_{A3}C_{A1}$ ) y luego se compara con un umbral ajustable manualmente por un comparador analógico LM211. Dos conectores jumper, llamados  $J_{A4}$  y  $J_{A5}$ , se usan para seleccionar el tipo de comparación (alarma en caso de que la entrada esté por encima del valor umbral, o bien, alarma en caso de que la entrada esté por debajo del umbral). Cuando los pines 1 y 2 de  $J_{A4}$  y los pines 2 y 3 de  $J_{A5}$ , se cortocircuitan, entonces el umbral es un límite inferior. Cuando los pines 2 y 3 de  $J_{A4}$  y los pines 1 y 2 de  $J_{A5}$  se cortocircuitan, entonces el umbral es un límite superior.

El valor de del umbral se genera con la ayuda de la referencia de tensión de 10 V de alta precisión AD587 conectada a un potenciómetro externo a través del conector  $J_{A3}$ . El AD587 genera una referencia de 10 V y así empleando un potenciómetro (multivuelta) de 10 k $\Omega$ , es posible regular el umbral  $t_A$  entre 0 y 10 V. La placa se alimenta a 24 V que son utilizados para generar cuatro conjuntos de voltajes independientes ( $\pm 15$  V cada uno) a través de cuatro reguladores de voltaje RS-2415. Así, los cuatro canales están eléctricamente aislados. También dispone de resistencias y diodos para evitar sobretensiones y que se pueda dañar el comparador.



**Figura 3.6** Comparador de umbral del sistema de interlocks rápidos



### **3.2.1 Mejoras introducidas en el comparador de umbral**

Inicialmente se esperaba que el módulo Fast Interlocks recibiera señales  $V_{\text{beam}}$  e  $I_{\text{beam}}$  (tensión y corriente del modulador, respectivamente) de valor positivo (entre 0 y 10 V). Sin embargo, finalmente el modulador, del fabricante JEMA, envía estas señales proporcionales a la tensión y corriente del haz del klystron negativas. La muestra de  $V_{\text{beam}}$  es de 1 V/15 kV y la de  $I_{\text{beam}}$  es de 1V/10 Amp (en ambos casos negativas). El modo de funcionamiento que se necesita del comparador es que cuando los valores de  $V_{\text{beam}}$  o  $I_{\text{beam}}$  sean más negativos que el umbral (asimismo negativo), entonces indique fallo. Hubo que realizar una serie de cambios en ciertos elementos respecto del esquema original, para conseguir la comparación con umbral negativo. Los cambios que se llevaron a cabo son:

- El circuito de protección de entrada debe conectarse a alimentación negativa -15 V.
- Los diodos 1N4148 deben colocarse en posición invertida.
- La tensión de referencia VREF debe ser -10 V, para lo cual hay que modificar ligeramente los componentes auxiliares del AD587.
- Deben intercambiarse las conexiones de las entradas i+ e i- (por medio de los jumpers).

En la figura siguiente se muestra el esquema circuital del comparador de umbral negativo del sistema fast interlocks de ESS Bilbao, que incorpora los cambios realizados.

Para verificar el correcto funcionamiento se simuló el esquemático obteniendo los siguientes resultados de simulación con LTSpice® de Analog Devices. En la figura 3.8 se muestra cómo cuando la señal de entrada es más negativa que el umbral (fijado en -5 V), la señal de salida V(fault\_out) pasa a nivel bajo, indicando fallo.

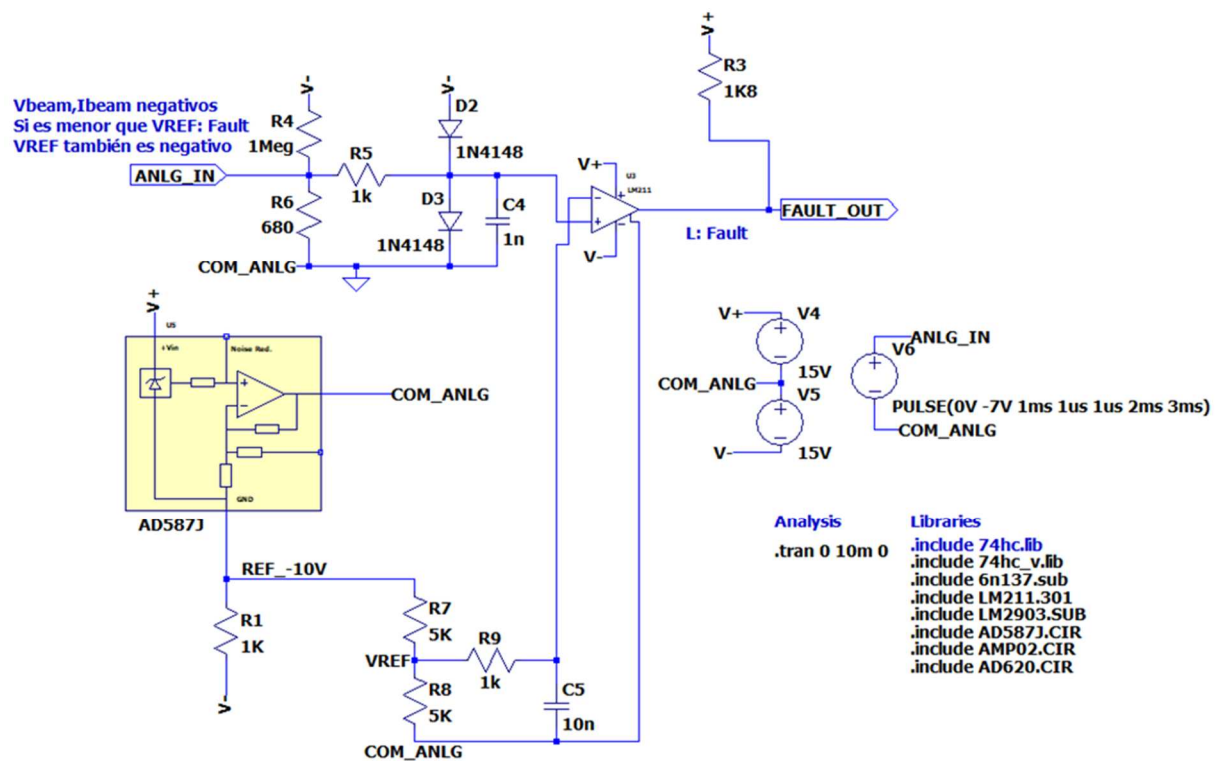


Figura 3.7 Esquemático en LTSpice del comparador con umbral negativo

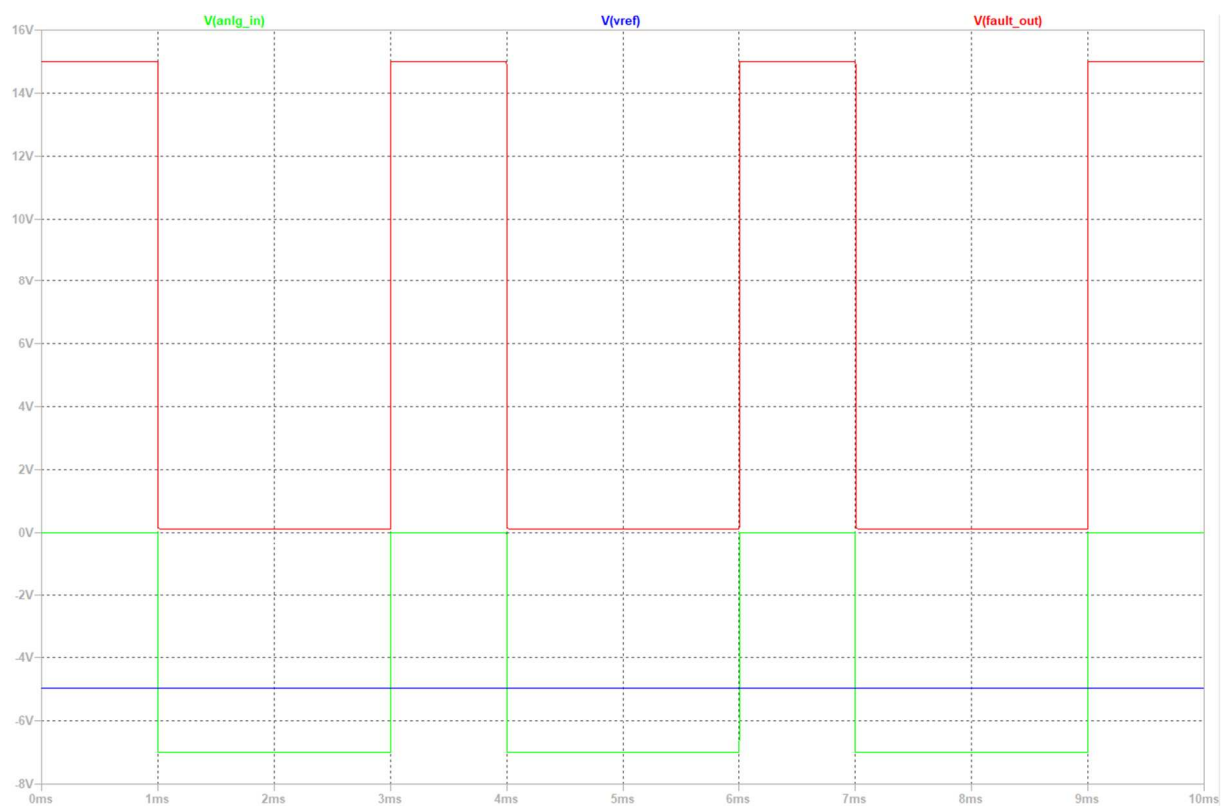


Figura 3.8 Simulación del esquema comparador de umbral negativo

### 3.3 Comparador de potencia del sistema Fast Interlocks

En este apartado se documentan las modificaciones realizadas en el comparador de potencia de RF del sistema de Interlocks rápidos del RFTS (Test Stand de RF de alta potencia). El objetivo de estas mejoras era evitar la indicación de falsas alarmas por bajas pérdidas de retorno, debidas a la falta de sincronización con la señal de disparo a causa de la lentitud del amplificador de instrumentación.

La función principal del comparador de potencia de RF en el sistema *FastInterlocks* [8] es contrastar la relación entre la potencia de la señal incidente y la potencia de señal reflejada con un umbral configurable, a través del cual interesará obtener altos niveles de pérdidas de retorno, típicamente superiores a 15 o 18 dB. En caso contrario, se puede reflejar en exceso la potencia hacia atrás y dañar el amplificador u otros elementos de la cadena.

Las entradas del comparador de potencia proceden de dos detectores de potencia ZX47-40+ de *Minicircuits* [9] que proporcionan una salida de tensión correspondiente a la potencia de RF presente a su entrada.

La expresión que relaciona la tensión de salida del detector con la potencia de entrada es:

$$V_{det} = 2 - 0.025 \bullet (P_{dBm} + 40) \quad (3.1)$$

donde:

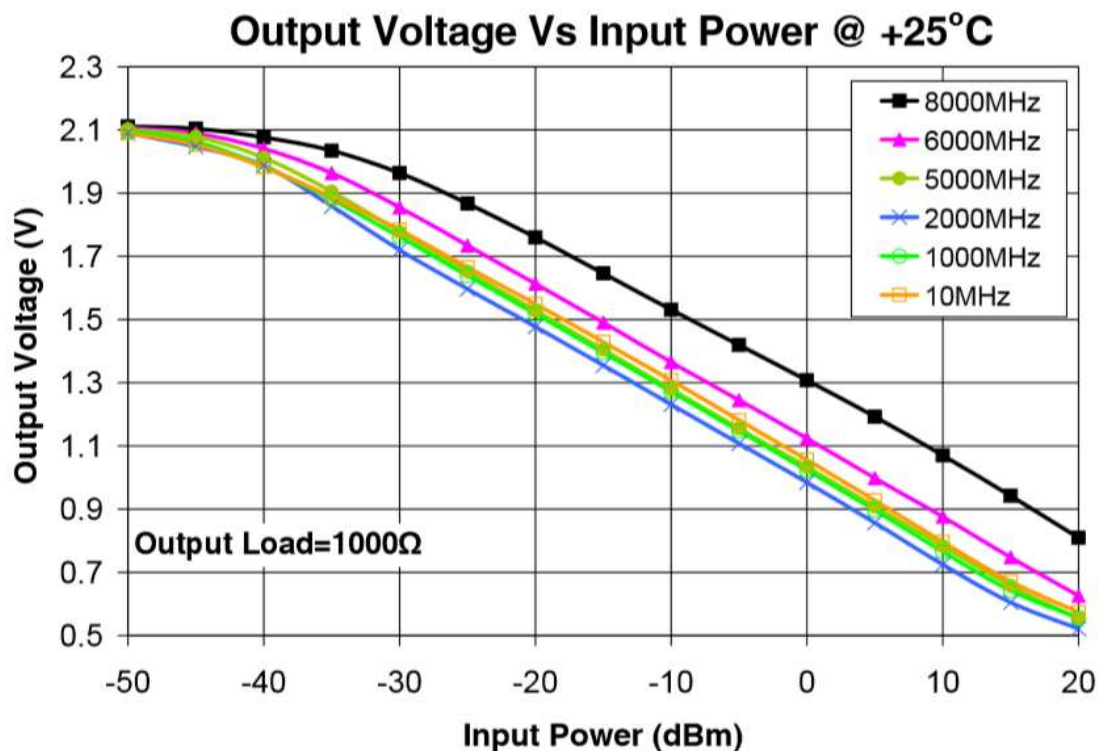
$V_{det} \equiv$  Tensión de salida del detector (V)

$P_{dBm} \equiv$  Potencia de entrada en el detector (dBm)

De la ecuación (3.1) se obtiene:

$$V_{det} = 1 - \frac{P_{dBm}}{40} \quad (3.2)$$

Dando valores a 3.2 se obtienen las siguientes gráficas variando la frecuencia de operación y a una temperatura de 25 °C.



**Figura 3.9** Voltaje de salida en función de la potencia de entrada al detector

Para comprender mejor el funcionamiento de la tensión que ofrece el detector de potencia de Minicircuits se presentan algunos valores recogidos en la siguiente tabla 3.1:

**Tabla 3.1** Valores de potencia en el detector y tensiones a la salida.

Potencia de entrada al detector (dBm)	Tensión de salida en el detector (V)
-40	2.0
-30	1.75
-20	1.5
-10	1.25
0	1.0

En resumen, para una potencia de -40 dBm o inferior, el detector de potencia proporciona una salida de 2 V. A medida que aumenta la potencia, la tensión de salida disminuye, con una pendiente de -0.025 V/dB.

A partir de (3.2), despejando la  $P_{dBm}$  obtenemos:

$$P_{dBm} = 40 \cdot (1 - V_{det}) \quad (3.3)$$

Definiendo la potencia incidente como  $P_{fwd}$ , potencia reflejada como  $P_{rev}$ , tensión del detector de la potencia incidente como  $V_{det}^{fwd}$  y tensión del detector de la potencia reflejada como  $V_{det}^{rev}$ , se puede definir las pérdidas de retorno  $RL$  como:

$$RL = P_{fwd} - P_{rev} = 40 \cdot (1 - V_{det}^{fwd}) - 40 \cdot (1 - V_{det}^{rev}) \quad (3.4)$$

Donde operando se obtiene:

$$RL = 40 \cdot (V_{det}^{rev} - V_{det}^{fwd}) \quad (3.5)$$

Para realizar esta función se emplea un amplificador de instrumentación AD620 [10], configurado con una ganancia de  $G = 4$ . Se decidió un valor de 4 en lugar de 40, a fin de manejar valores de tensión más convenientes, entre 0 y 10 V. Asimismo, el umbral de comparación se configura a través de una referencia de tensión de 10 V (proporcionada por un AD587) y un potenciómetro de 10 K $\Omega$  multi-vuelta que dispone de una ruleta de 000 a 999, en la que se establece el valor del umbral de detección que sea requerido.

Aludiendo a los valores que se toman de ejemplo en la siguiente tabla 3.2, se puede comprender cómo se ajustarían los valores de dicho umbral.

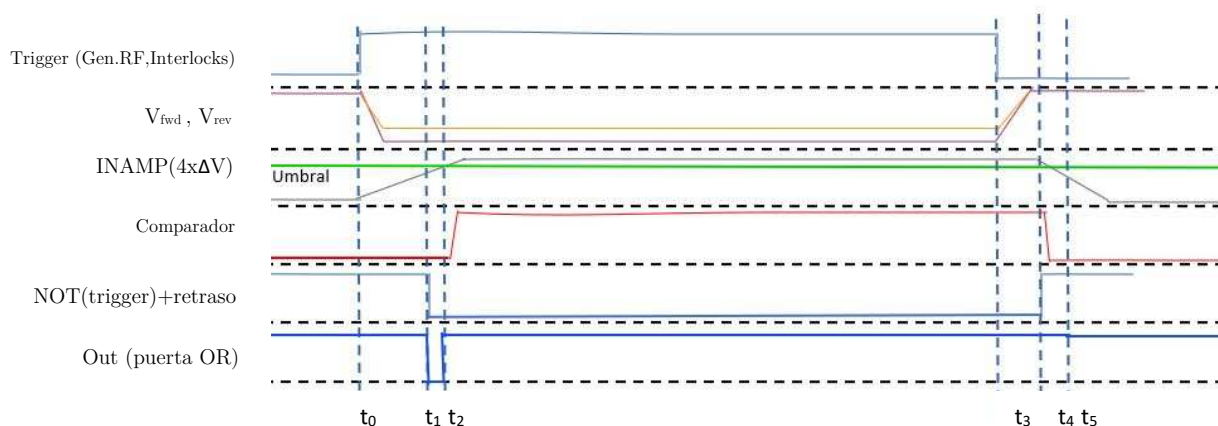
**Tabla 3.2** Valor del umbral en función de la ruleta del potenciómetro multi-vuelta

Valor de la ruleta	Umbral de Tensión (V)	Umbral de pérdidas de retorno (dB)
100	1.0	10
150	1.5	15
180	1.8	18
200	2.0	20

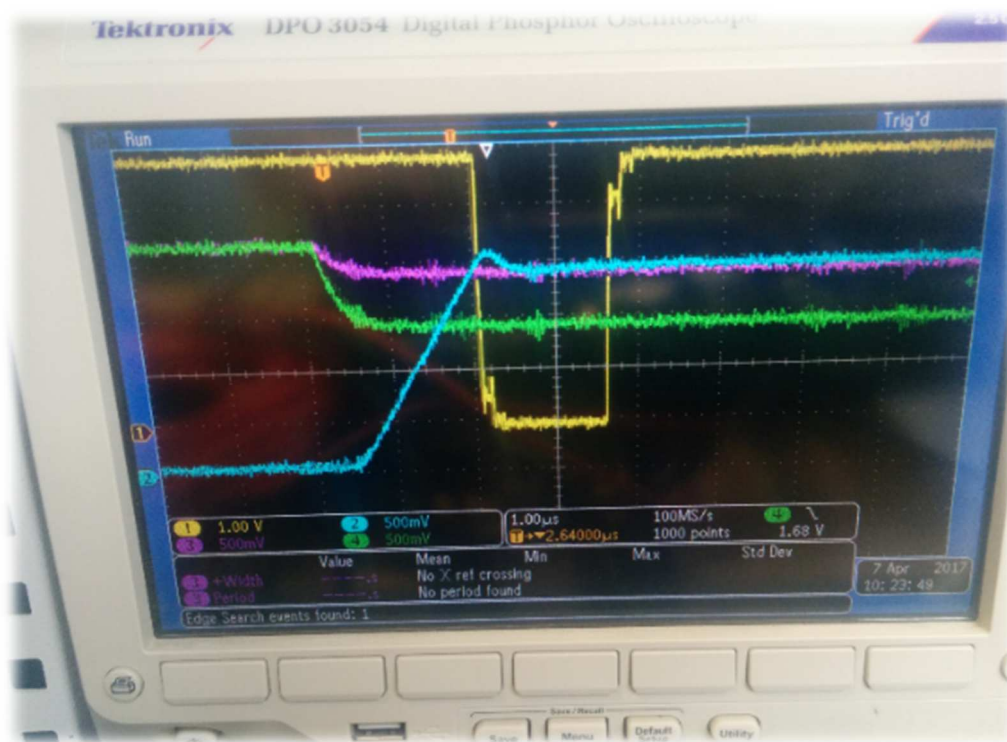
La comparación de potencia de RF incidente y reflejada sólo tiene sentido durante el tiempo de duración del pulso, cuyo inicio lo marca una señal de disparo (trigger). Fuera del pulso, dicha comparación no tiene relevancia y además el sistema mostraría un fallo de forma incorrecta, desencadenando una falsa alarma en el componente Fast Interlocks e interrumpiendo el proceso de funcionamiento de los sistemas de RF.

Cuando se realizó el montaje del comparador de potencia, se observó en las primeras pruebas de laboratorio que la operación era correcta para señales de onda continua y pulsadas. Sin embargo, una vez instalado en el RF Test Stand, con el sistema de sincronismo de las señales de RF pulsadas (con anchos de pulso entre 0,5 y 3,5 ms y frecuencias de repetición entre 1 y 14 Hz) y los detectores de potencia empleados, aparecía al inicio del pulso de RF una indicación de fallo por exceso de pérdidas de retorno en la señal de salida ‘Out(puerta OR)’ durante un instante  $t_2-t_1$ . Este hecho se representa en el diagrama de formas de onda de la Figura 3.10.

En la Figura 3.11 se presentan las señales de tensión del detector de potencia del klystron incidente (en verde), tensión del detector de potencia del klystron reflejada (en rosa), tensión a la salida del amplificador de instrumentación (en azul) y tensión de salida del comparador de umbral (en amarillo). En torno a los 4  $\mu$ s (siendo de 1  $\mu$ s/división la escala temporal) la señal amarilla pasa a nivel bajo momentáneamente, indicando una falsa alarma, debida a la lentitud de la tensión de salida del amplificador de instrumentación en superar el umbral fijado, restringida por el slew rate del AD620.

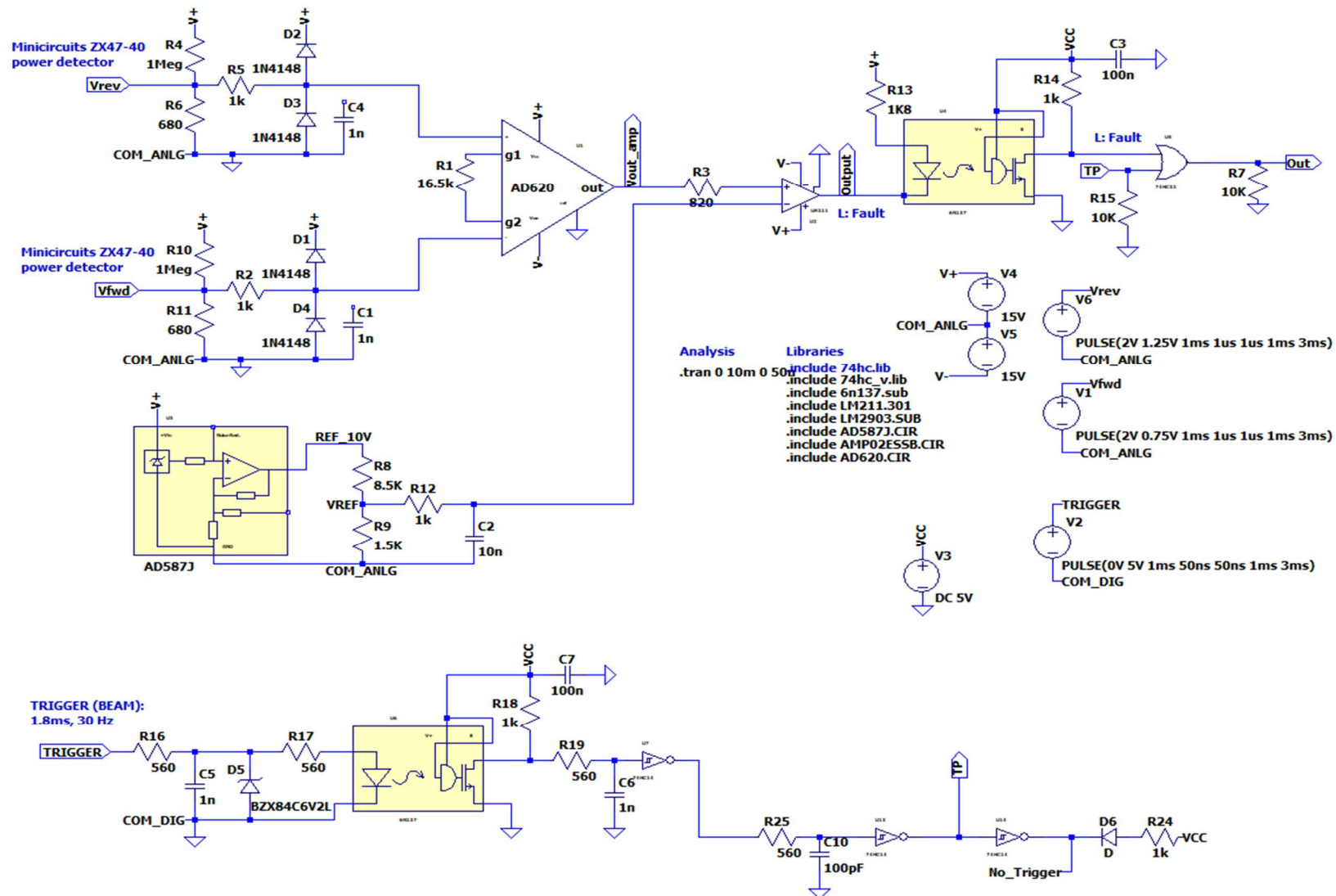


**Figura 3.10** Diagrama temporal con el amplificador de instrumentación AD620



**Figura 3.11** Representación de la falsa alarma en el osciloscopio

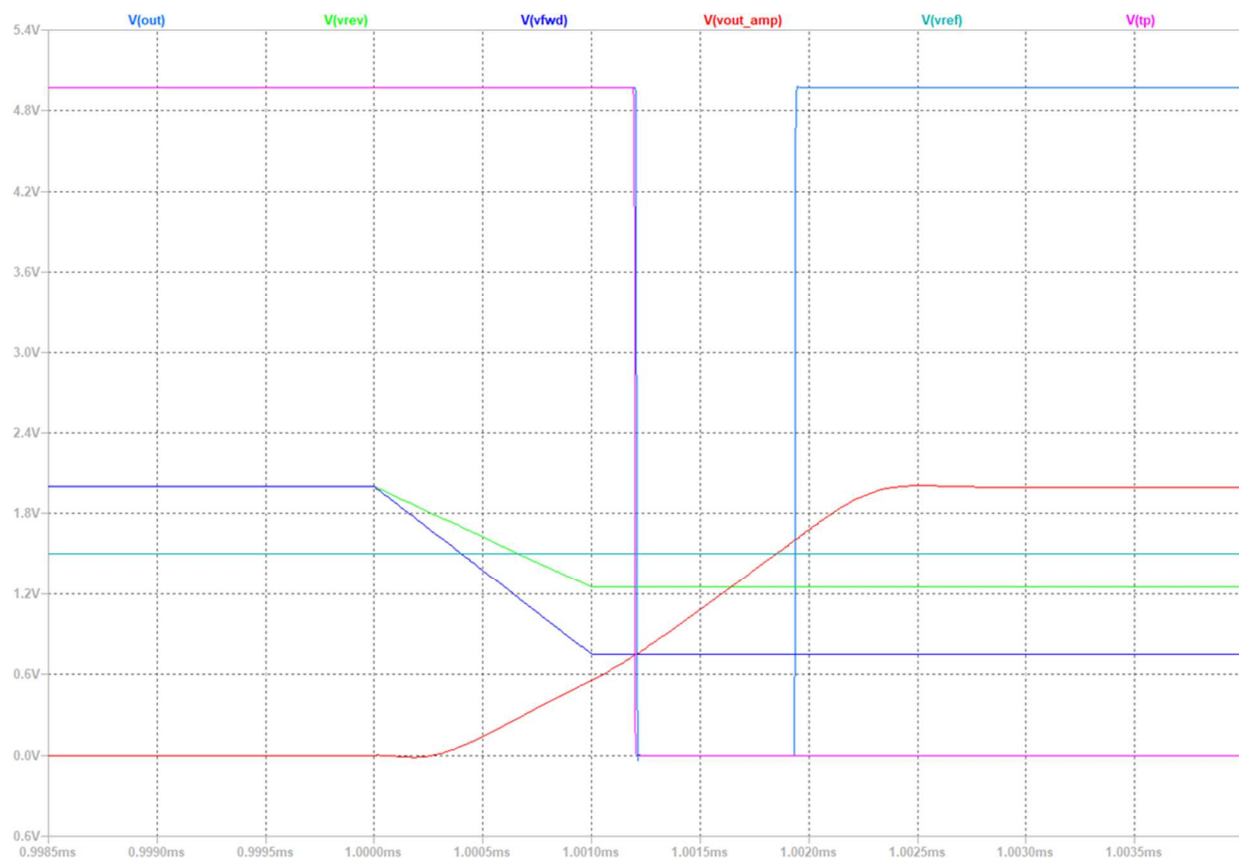
Una vez introducido el problema en cuestión se simuló el proceso, el cual se llevó a cabo con el programa LTSpice, a continuación se presentan el esquemático y el resultado de la simulación.



**Figura 3.12** Esquemático del comparador de potencia



Como se pudo comprobar en el resultado de la simulación, el falso error estaba presente en la señal de salida del circuito “V(out)”.



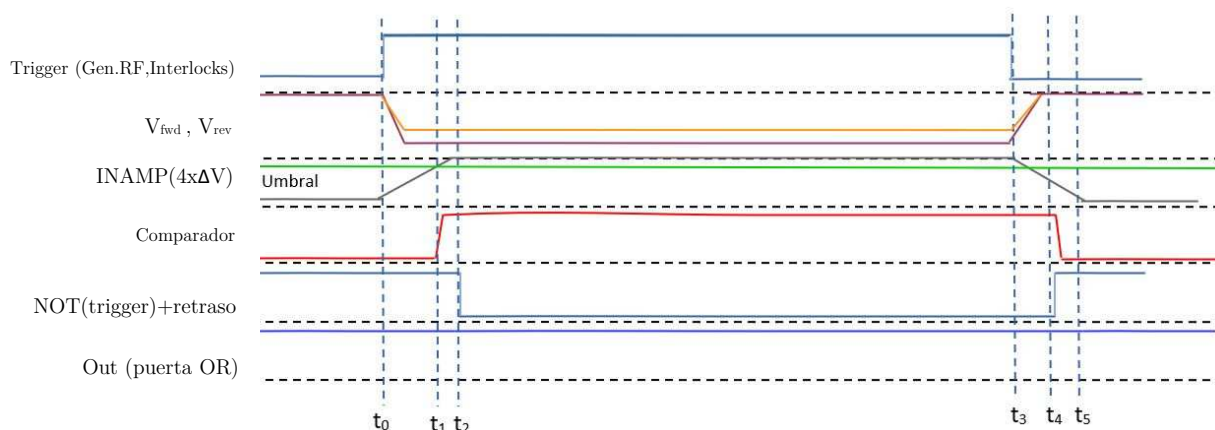
**Figura 3.13** Señales obtenidas de la simulación en LTSpice con el falso error, empleando el amplificador de instrumentación AD620

### 3.3.1 Sustitución del amplificador de instrumentación AD620

Se pudo comprobar que el amplificador de instrumentación AD620 presentaba un Slew Rate de  $1.2 \text{ V}/\mu\text{s}$  [11] de valor típico, insuficiente para realizar la transición de valor nulo a cuatro veces la diferencia de tensión entre la tensión de la señal reflejada y de la tensión de la señal incidente.

Para conseguir resolverlo se decidió sustituir el componente AD620 por un amplificador de instrumentación más rápido, pero que a la vez fuera compatible eléctrica y mecánicamente (función, pines, alimentación, encapsulado DIP-8), para evitar realizar cambios importantes en la tarjeta. El componente elegido fue el amplificador AMP02 [11], con un Slew Rate de  $4\text{-}6 \text{ V}/\mu\text{s}$ , lo que permite una respuesta más rápida y evita que se origine el fallo al inicio del pulso.

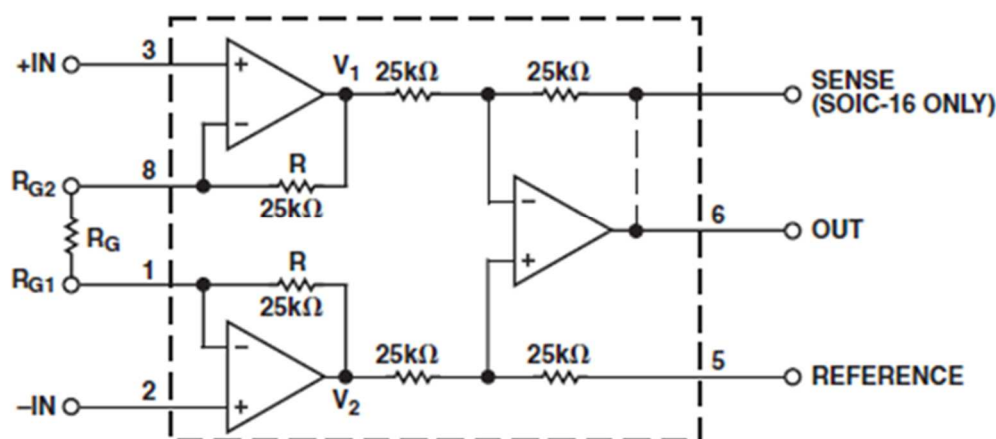
En el siguiente diagrama temporal de señales, se demuestra que ya no aparece el falso error en la señal ‘Out(puerta OR)’, utilizando el amplificador de instrumentación AMP02 en lugar del AD620.



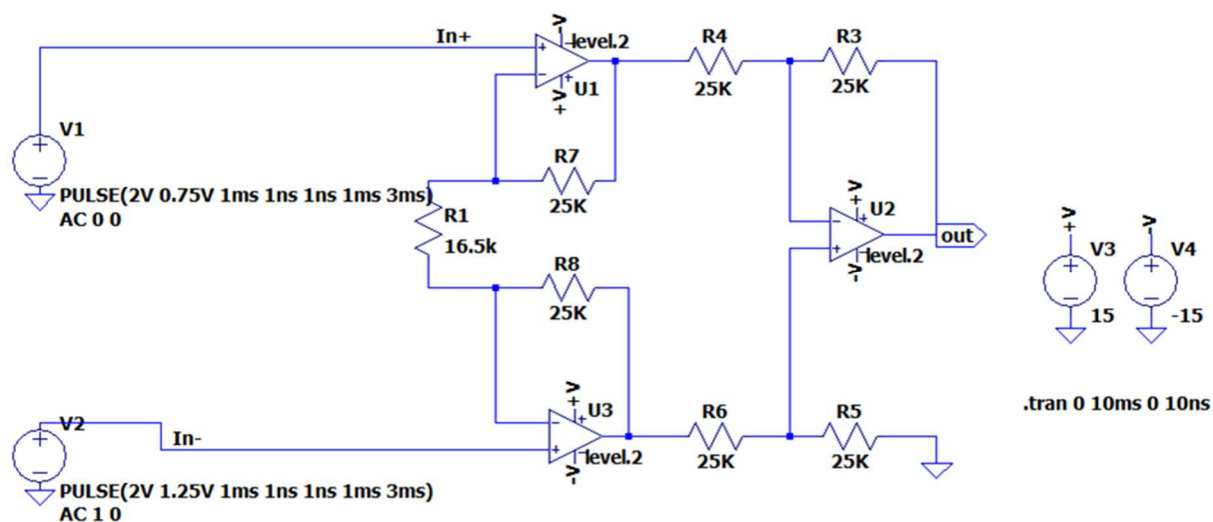
**Figura 3.14** Diagrama temporal con el amplificador de instrumentación AMP02

Como no se disponía de un modelo Spice adecuado del AMP02 para realizar las simulaciones incluyendo las características temporales, hubo que crear un modelo comportamental del componente, a partir de la hoja de características y de su equivalente circuital que proporcionaba el fabricante.

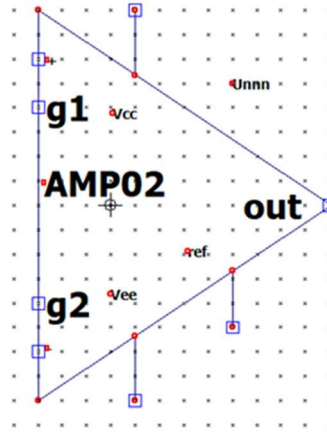
El modelo del AMP02 creado está basado en un amplificador de instrumentación de tres amplificadores operacionales ‘level 2 universal OPAMP’ de la librería de LTSpice, ajustando los parámetros de ‘Slew-Rate’ con valor 6 V/ $\mu$ s y GBW (ganancia por ancho de banda) de 3.5 MHz, según la hoja de datos del fabricante del AMP02. A continuación, se representan los elementos que componen internamente el AMP02 y su correspondiente esquemático.



**Figura 3.15** Esquema interno del amplificador de instrumentación AMP02 de Analog Devices



**Figura 3.16** Esquemático del AMP02 realizado en LTSpice de Analog Devices



**Figura 3.17** Símbolo del componente de librería del amplificador de instrumentación AMP02

Para configurar una ganancia  $G = 4$  del circuito con el AMP02, se calculó la resistencia  $R_G$  correspondiente de  $16,5 \text{ K}\Omega$ .

Una vez se había incluido el modelo en la librería del LTSpice, se simuló el esquemático inicial reemplazando el amplificador de instrumentación AD620, por el nuevo amplificador AMP02.

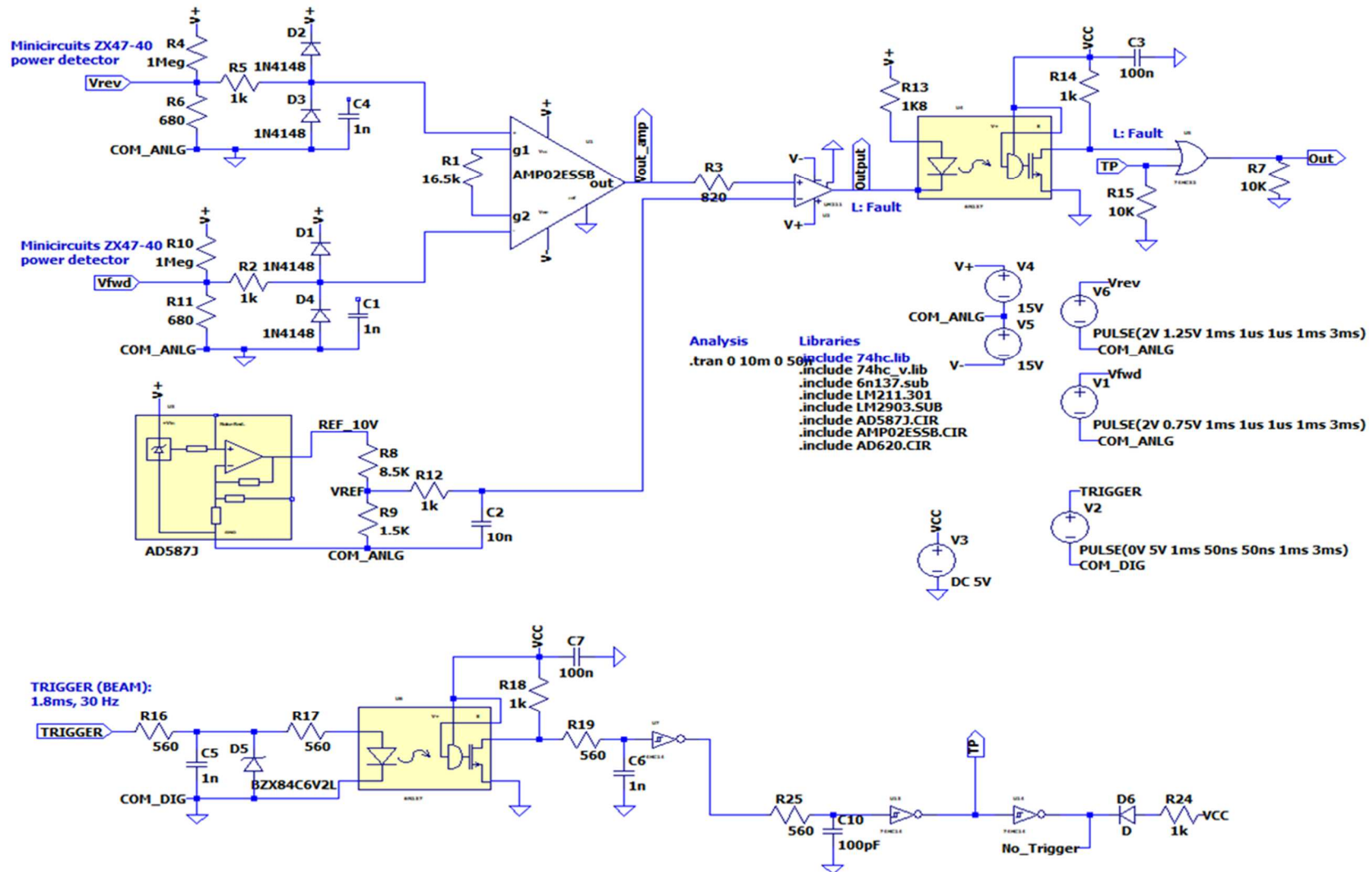
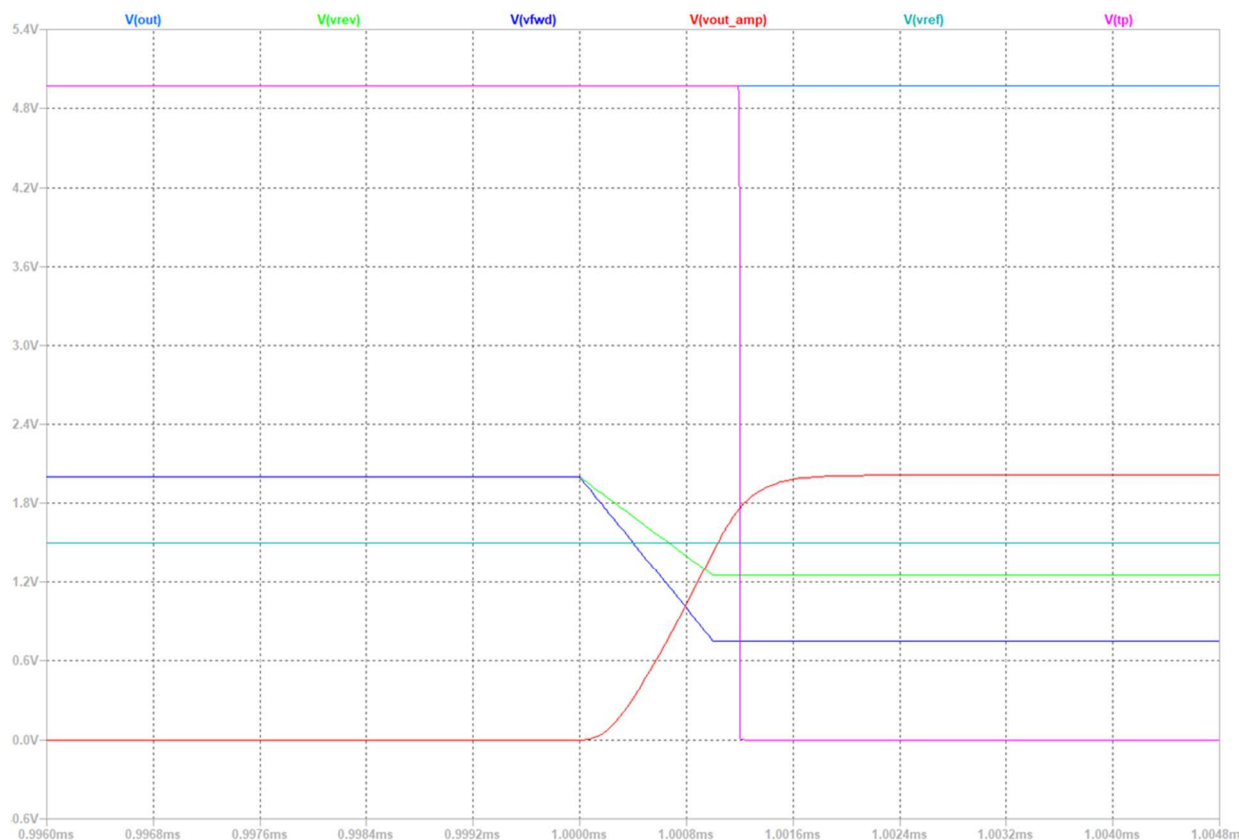


Figura 3.18 Esquemático del comparador de potencia con el amplificador de instrumentación AMP02 (ESS Bilbao)

Se aprecia en la Figura 3.19 cómo el valor de la salida del circuito AMP02 alcanza el umbral en  $0,95 \mu s$  (frente a los  $1,8 \mu s$  que tarda el circuito basado en AD620), así como los retrasos de los componentes que conforman el trigger. La pendiente de la respuesta en la salida 'V(vout\_amp)', en el caso del amplificador de instrumentación AMP02 es más alta, por lo tanto, queda anulado el efecto de falsa alarma por potencia reflejada (nivel bajo en la señal Vout), cuando se activa la comparación.



**Figura 3.19** Señales obtenidas en LTSpice con el amplificador de instrumentación AMP02

Realizadas las citadas mejoras del sistema de enclavamientos rápidos, fue posible operar, de forma segura y sin interrupciones, el klystron de potencia en la instalación de pruebas de RF del Consorcio ESS Bilbao.

## **Capítulo 4: Interfaz gráfica en MATLAB para adquisición de medidas automáticas del klystron**

La realización de las campañas de medidas en el reacondicionamiento de los klystron requería un método automatizado, de tal manera que el usuario sólo se tuviera que preocupar de tomar la medida mediante una única interacción con el osciloscopio, evitando tener que realizar diferentes calibraciones manuales previas a la toma de datos. Con ese propósito, y el de minimizar las tareas a ejecutar por el usuario, empleando menos tiempo en cada adquisición de las medidas, se optó por la realización de un software, capaz de agilizar el procedimiento. La opción que se escogió, fue la de una interfaz gráfica de usuario embebida en el osciloscopio mediante MATLAB® de MathWorks.

De esta forma, se realizó una GUI (Graphical User Interface) en MATLAB con el fin de automatizar un proceso de medidas con el osciloscopio de TekTronix (DPO7354), cuyo software está instalado en el propio osciloscopio, ya que éste dispone de un sistema operativo Windows XP. Con ello se consiguió facilitar al usuario de la instrumentación de medida, la tarea de adquirir y también de analizar medidas relativas al klystron.

La medida de alta potencia de RF del klystron es una medida muy importante en la caracterización del dispositivo. Ésta se realiza en función del punto de operación, es decir, según las condiciones de tensión y corriente proporcionadas por el modulador, así como la duración del pulso de alta tensión, la potencia de generador de RF, la duración del pulso de RF, la frecuencia de repetición de pulsos y la frecuencia de trabajo. Se necesita conocer y adquirir medidas de cuáles son las tensiones en los detectores de las potencias incidente y reflejada del klystron, la potencia a la salida del amplificador driver de RF, para obtener resultados de cómo se comporta todo el sistema y realizar los ajustes que sean necesarios en todo momento. Gracias a la interfaz gráfica de usuario, dicha tarea se presenta más favorable y práctica, permitiéndose también guardar los resultados que serán exportados e interpretados con la herramienta de hoja de cálculo EXCEL® de Microsoft.

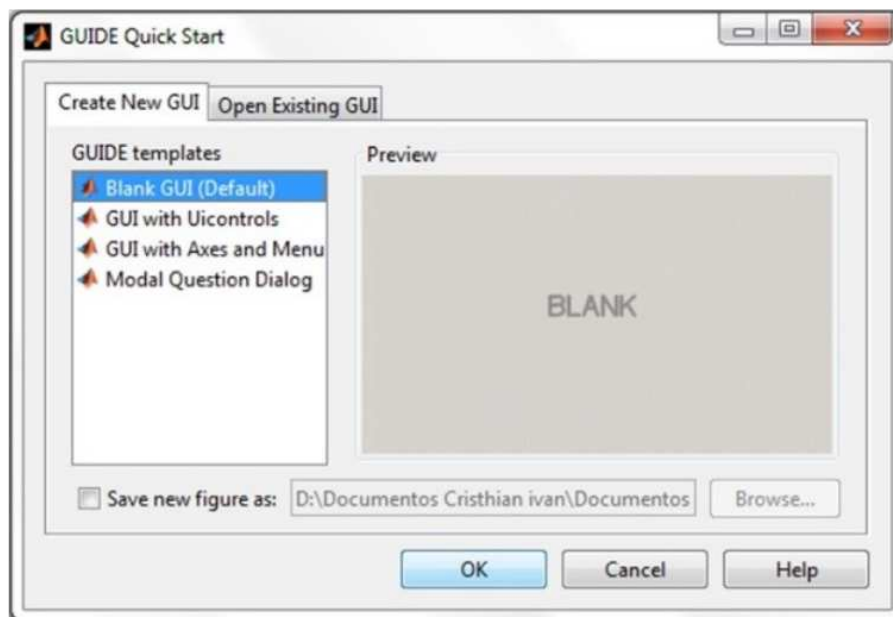
## 4.1 Interfaz gráfica de usuario en MATLAB para medidas con el klystron

La GUI consta de una serie de elementos que proporcionan una estructura compuesta por la combinación de botones, cajas de texto, tablas, y ejes de representación de gráfica, que permiten mejor visualización de los resultados.

En primer lugar, se genera un script que describe las funciones que intervienen en el GUI correspondientes a los elementos que lo componen. La GUI se compone de dos partes, una es el script con extensión '.m', y la otra, que es con la que el usuario interactúa, tiene extensión '.fig', ambas deben estar alojadas en el mismo directorio, para su correcto funcionamiento.

Para detallar de una forma más visual cómo se realizó la GUI, se presentará una imagen del entorno de usuario, y se explicará paso a paso, el diseño y la programación de las funciones.

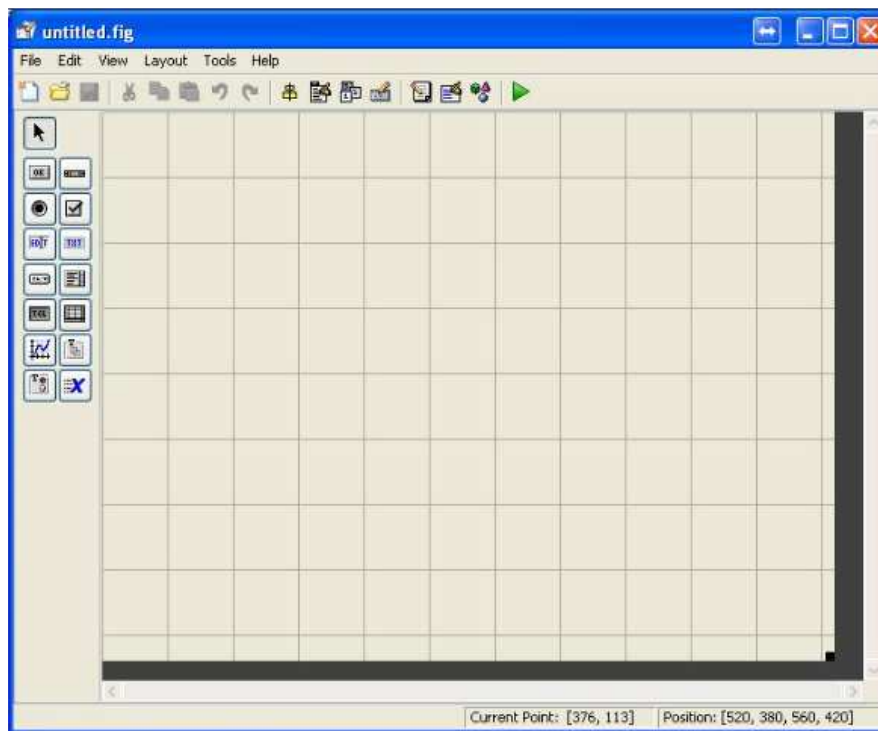
Para abrir este entorno, basta con ejecutar en la línea de comandos de MATLAB, la siguiente instrucción ('guide'), se abrirá un menú como el siguiente:



**Figura 4.1** Apertura de un nuevo GUI



Una vez pulsamos en ‘Blank GUI (Default)’, se muestra el entorno donde realizaremos las modificaciones de los objetos que se van a incluir en nuestra interfaz.



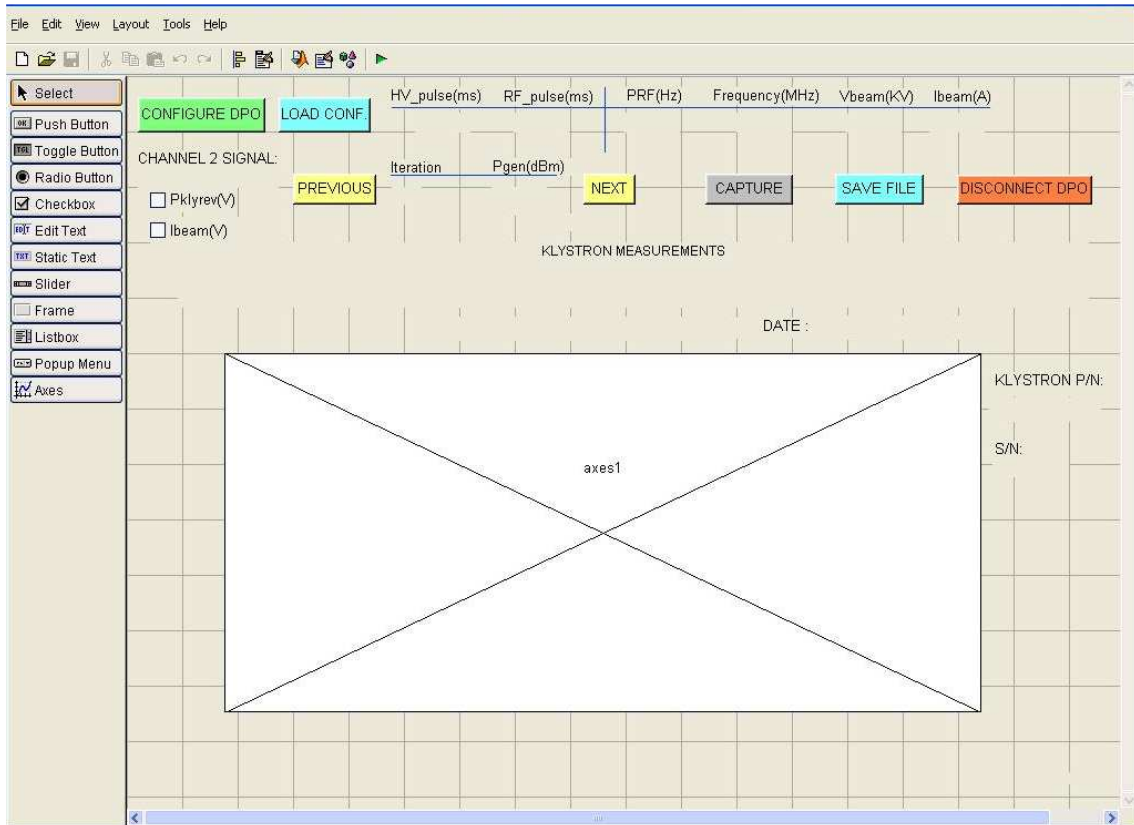
**Figura 4.2** Entorno de diseño de los objetos

En primer lugar, se añadieron al entorno los botones (‘push buttons’), que servirán para accionar las funciones asignadas a la pulsación de cada uno de ellos.

En segundo lugar, se añaden las cajas de texto (‘static text’), para representar en ellas los valores de los parámetros que queramos visualizar. Se pueden añadir títulos también con estos elementos.

También se necesita unos ejes para la representación de las gráficas que se van realizando a medida que adquirimos datos, estas ayudan a tener una referencia más visual.

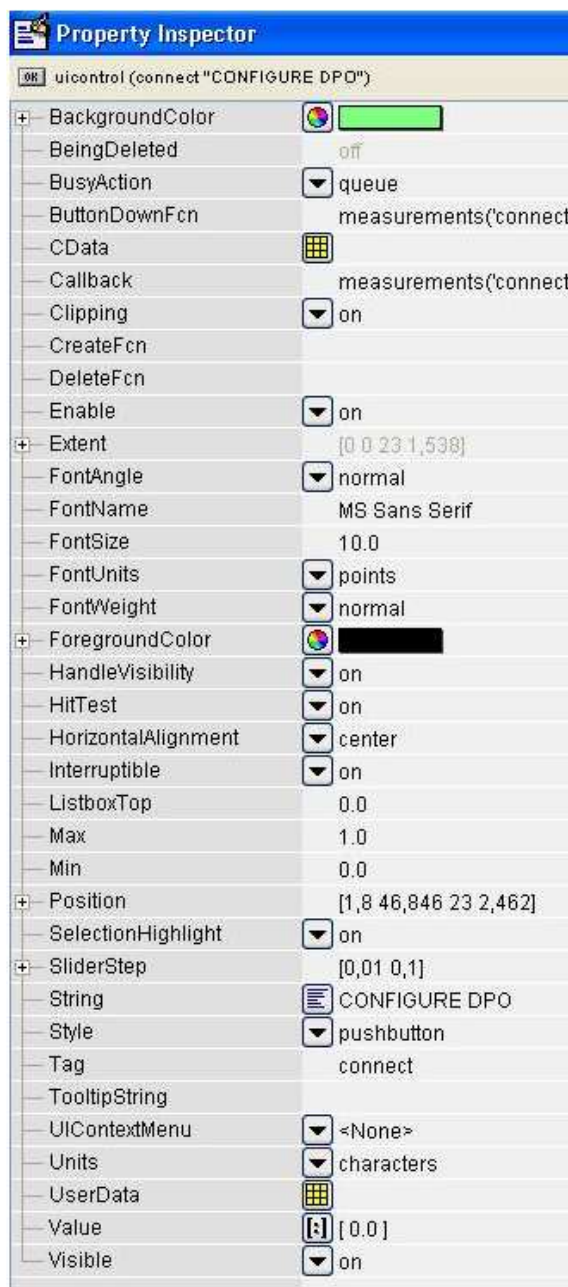
A continuación, se mostrará lo que es el resultado de la combinación de los elementos seleccionados en la GUI, como ya se ha estado comentando previamente:



**Figura 4.3** Vista del entorno de diseño de la GUI


Se representaron los objetos que van a conformar la GUI, se modificaron y se configuraron de manera personalizada, para añadirlos al entorno, sencillamente pulsando en la parte izquierda de la ventana en la lista de objetos y colocándolos en el lugar que se desee.

Una vez emplazados, se configuran en función de las preferencias del diseñados, mediante las opciones que se presentan pulsando en el botón derecho sobre ellos, y después en el menú que se despliega seleccionamos 'property inspector', a partir de ahí se pueden ir cambiando los atributos de cada objeto (el tamaño, el texto que queremos que muestre, la posición en el GUI, la etiqueta 'tag' con la que se identificará en el script, color de fondo y nombre de la función de 'callback', entre todas las propiedades que nos afectan).



**Figura 4.4** Menú desplegable de ‘property inspector’ de los objetos

Toda la configuración se puede llevar a cabo desde el script basándonos en comandos, pero el hecho de ir cambiando los objetos gráficamente, permite asegurarse de que se han configurado los atributos que se desean.

Se puede desplegar una lista con los objetos que hemos introducido en el GUI, si pulsamos en el icono de la parte superior , entonces se observa la siguiente lista desplegable:

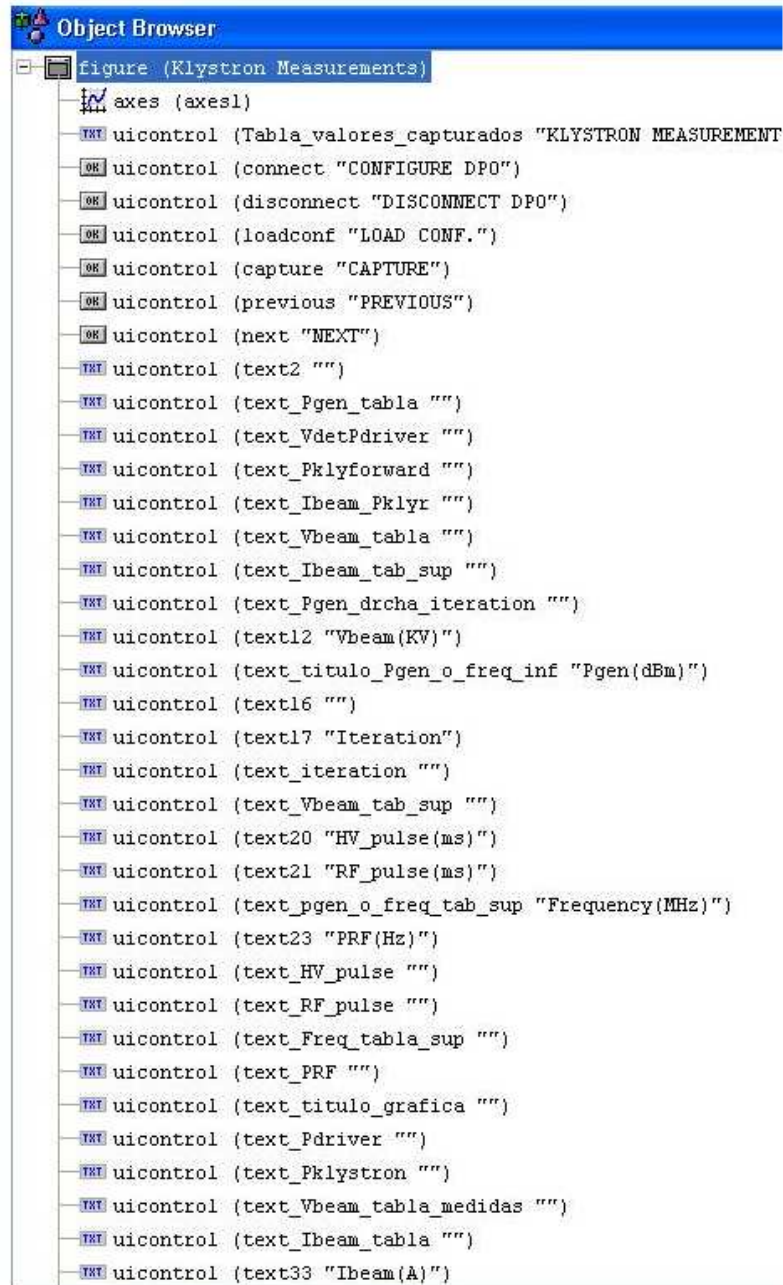


Figura 4.5 Lista de objetos en la GUI

Una vez ajustados los objetos, se procede a describir las funciones de ‘callback’ de los objetos, que intervienen en la GUI.

#### Capítulo 4: Interfaz gráfica en MATLAB para adquisición de medidas automáticas del klystron

Antes de entrar en detalle con las funciones de ‘callback’ se va a explicar cómo será el funcionamiento y la operación de la GUI, después se irá función a función para explicar su relevancia en el programa. Se parte para dicha explicación, de la siguiente imagen que representa el aspecto final de la GUI, cuando ésta sea ejecutada.

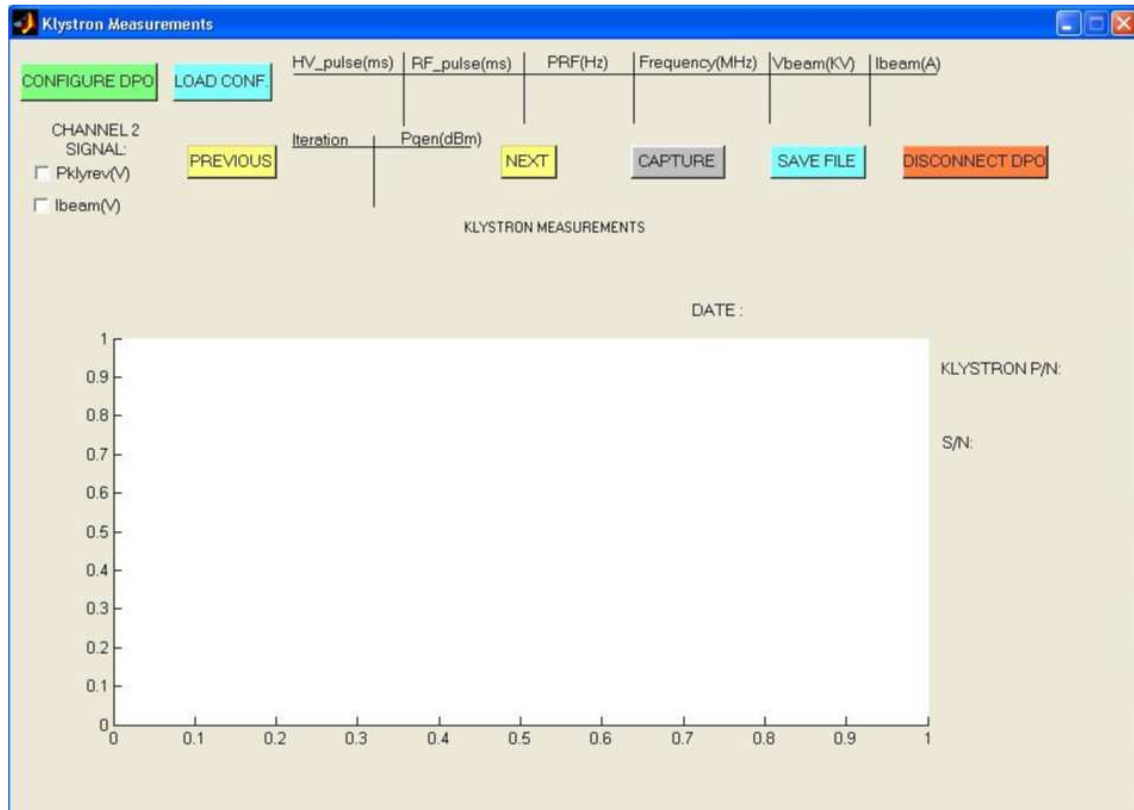


Figura 4.6 Aspecto final de la GUI

Para entender cuál es el funcionamiento de la GUI, y cómo se trabaja con ella, se explicará cualitativamente su operación, y cómo se interactúa con el programa.

Ante el inicio de la ejecución del programa, hay tres pasos previos fundamentales:

1. El primer paso es la configuración del osciloscopio, pulsando el botón verde ‘**CONFIGURE DPO**’, con ello se configuran los cuatro canales, escala temporal (eje horizontal), voltaje por división (eje vertical), la señal externa de disparo, modo de representación promediado ‘*Average Mode*’.

2. A continuación, se carga un archivo de configuración con **'LOAD CONF'**, el cual se encarga de cargar la configuración (settings) de la medida.
3. Después, seleccionamos cuál es la sonda que está conectada en el canal 2, correspondiente a una de las dos posibilidades, la señal **'Ibeam(V)'** o la señal **'Pklyrev(V)'** dichas señales están etiquetadas en las sondas correspondientes.

Para el siguiente paso se van capturando las medidas que son necesarias en ese momento. El proceso de la captura es el siguiente, se ejecuta el botón **'CAPTURE'** para tomar la primera captura, es importante asegurarse de haber cargado el fichero correctamente y de haber seleccionado la señal del canal 2 que hemos conectado. Se toma el valor con **'Capture'** y le damos a siguiente **'NEXT'**, nuevamente capturamos, así hasta que consideremos que tenemos las capturas de los datos que nos interesan. Si vemos que nos hemos equivocado o que el resultado no es el esperado, se puede retroceder con **'PREVIOUS'** al valor anterior y volverle a capturar. Así con todos los valores que necesitemos, que se encuentren dentro de las opciones de barrido, ya que existen dos tipos de medidas; las de potencia de salida frente a la potencia de entrada **'PinvsPout'**, y la medida de la potencia de salida en función de la frecuencia de operación **'Frequency Measurement'**.

Si nos encontramos en el primer tipo de medida, estaremos variando el valor de la potencia de generador de RF con los botones ya mencionados **'Previous'** y **'Next'**. Capturando el que esté configurado correspondientemente en cada momento. Si por el contrario el tipo de medida que se está realizando es frecuencial, entonces lo que se varía es la frecuencia de operación del klystron. Todas las medidas capturadas cuando hayamos finalizado se pueden guardar exportando los resultados a un archivo cuya extensión es **'csv'**, dicho archivo es interpretable por MATLAB y también por EXCEL, entre otros. Para realizar la exportación de resultados y el guardado de los mismos, es necesario pulsar el botón del GUI **'SAVE FILE'**, en ese momento se presentará el menú para guardar el documento en el directorio que seleccionemos, y con el nombre que introduzcamos, notar que el fichero por defecto tiene asignada la extensión **'csv'**. Si no le introducimos ninguna extensión, automáticamente se le atribuye **.csv**.

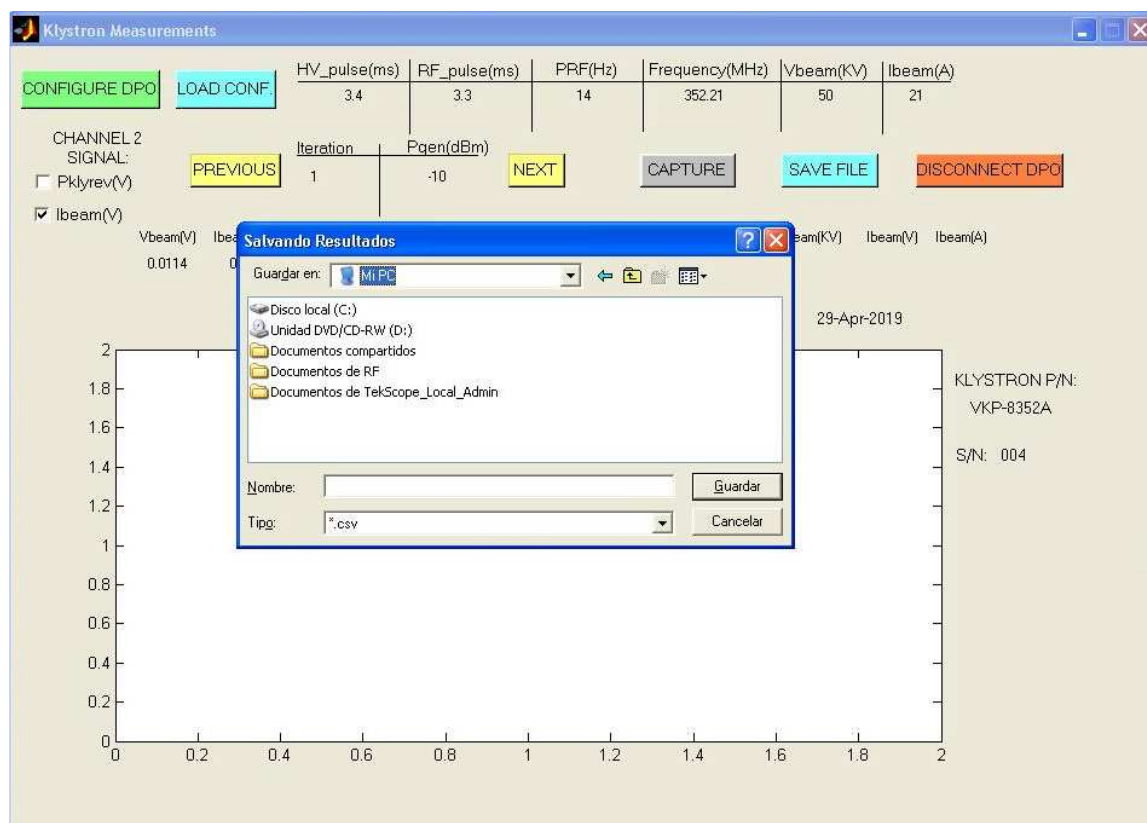


Figura 4.7 Salvando resultados en archivo .csv

Además de realizar la exportación del fichero a .csv, se guarda automáticamente una captura del entorno del programa para en un futuro acceder, a los datos de configuración y poder archivarlos en un historial. El proceso de representación de los resultados de las medidas con la interfaz de MATLAB y cómo procesarlos se adjunta en el ANEXO II.

Por defecto, el nombre que se le asigna a la imagen, cuya extensión es '.jpg', es 'Klystron\_Measurements\_#fecha('día'-'mes'-'año')\_#hora('horas'-'min.'-'segundos') .jpg'. Por ejemplo: 'Klystron\_Measurements\_11-Apr-2019\_16-38-38.jpg'.

En cuanto a la función del botón '**DISCONNECT DPO**', cabe destacar que está programado para cerrar el objeto que se creó del osciloscopio para todo el envío y recepción de comandos, se libera el espacio de las variables en la memoria y se sale de la aplicación.



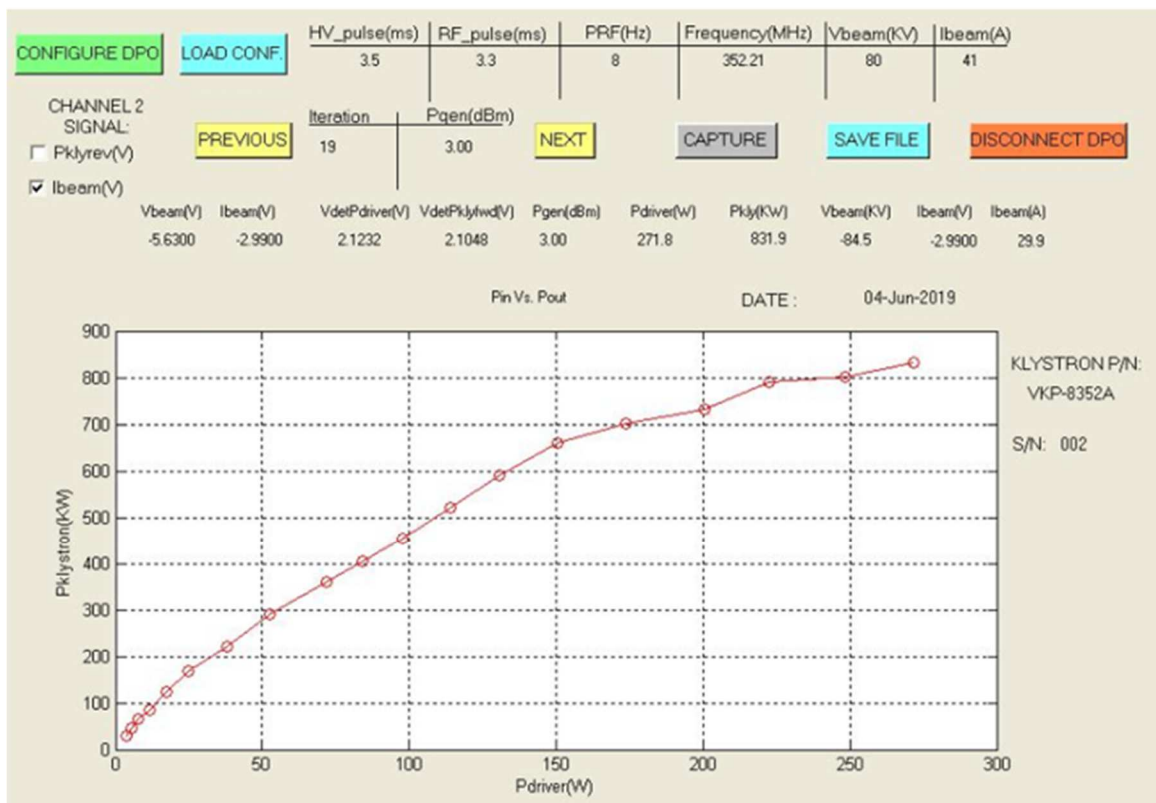


Figura 4.8 Ejemplo de imagen guardada de medida PinVsPout con extensión '.jpg'

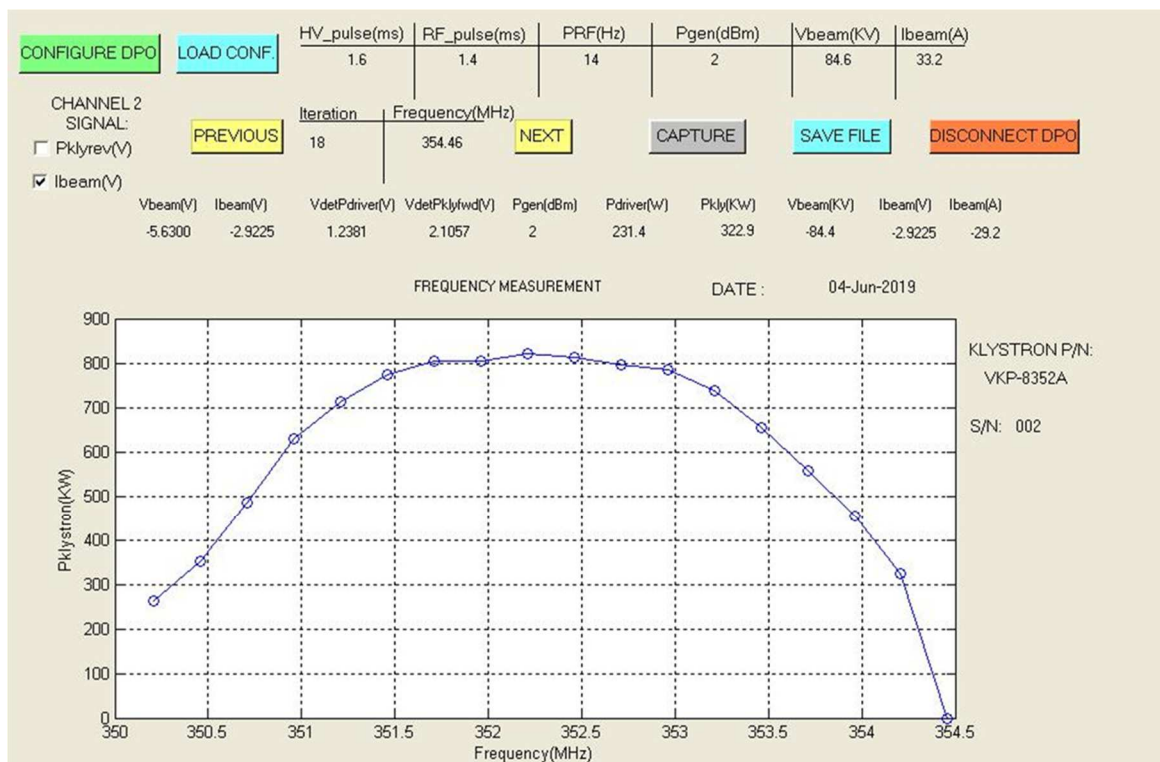


Figura 4.9 Ejemplo de imagen guardada de medida frecuencial con extensión '.jpg'



## 4.2 Estructura y código del GUI

En toda GUI hay una parte de entorno gráfico o figura de interfaz gráfica al usuario, y otra parte de programación o código ejecutable, que posibilita el funcionamiento deseado de los objetos del programa. La estructura de dicho código está explicada esquemáticamente a continuación.

El código está formado por una función principal que posee el mismo nombre que el archivo del script. Cada función de ‘Callback’ son funciones definidas localmente, dentro de la principal. También cabe destacar que se incluyen plantillas de funciones de ‘callback’ que se generan automáticamente. Se genera una función de inicialización, además de un ‘callback’ de salida. El diseñador modificará las funciones con cierto criterio, para que el GUI funcione como haya previsto. Por tanto, el programador, si lo cree conveniente, añadirá código o modificará las funciones de ‘callback’, y opcionalmente, podrá introducir comandos en la rutina de inicio, y de finalización, si así lo desea.

Antes de presentar el esquema de la GUI, hay que destacar que nos encontramos ante una estructura donde las variables que intercambian las funciones vienen definidas con la siguiente denominación ‘*handles.nombre\_de\_variable*’ lo que indica ‘*handles*’ es la estructura, si queremos modificar variables que pertenecen a la estructura los ‘*handles*’, debemos utilizar la función ‘*guidata(hObject,handles)*’ después de cada instrucción contra el objeto.

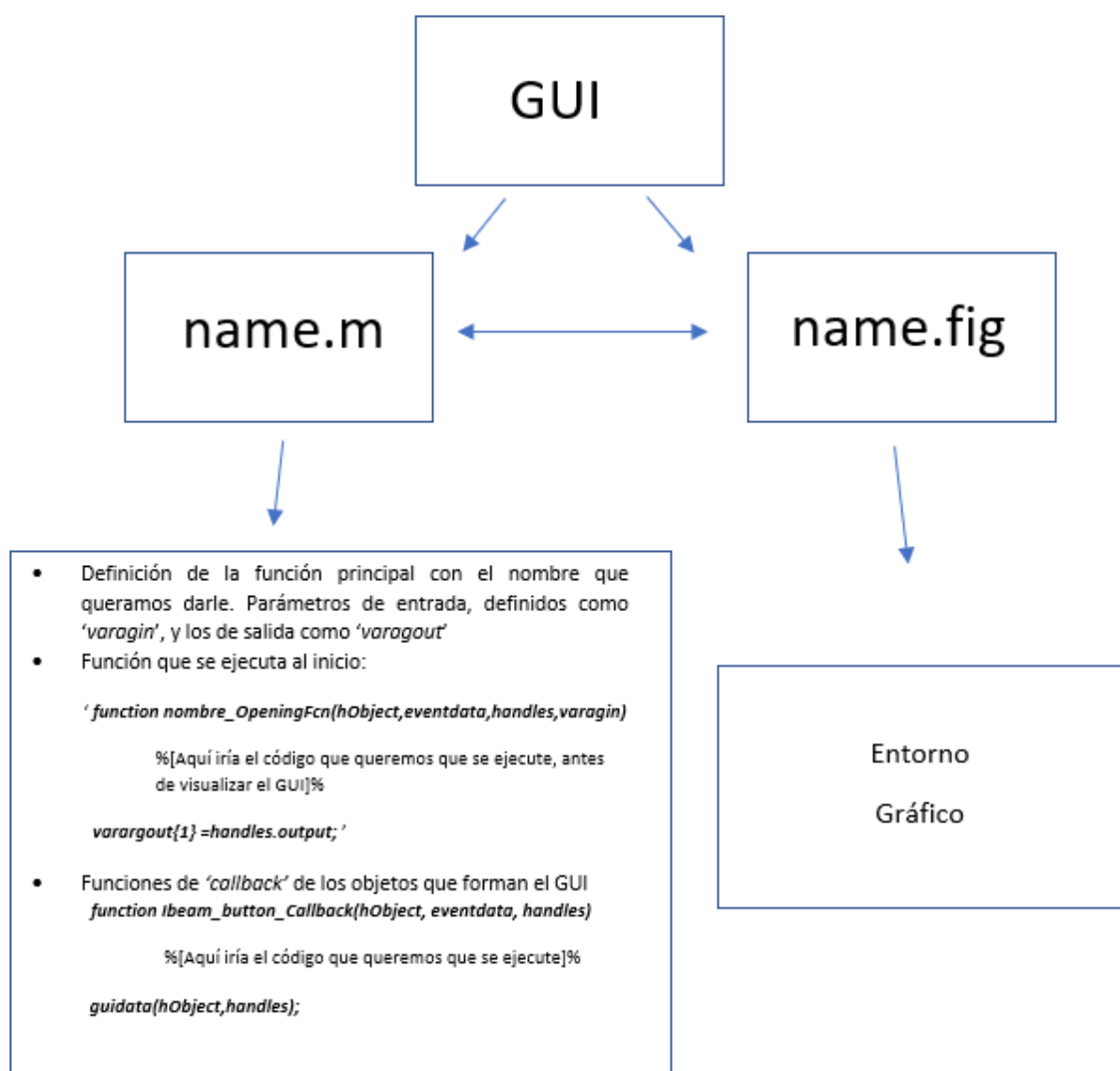


Figura 4.10 Estructura de la GUI para medidas automáticas con los klystron

### 4.3 Fichero de configuración

Los ficheros de configuración son esenciales para cargar los parámetros de las medidas que se van a realizar con la GUI, en función de los parámetros del klystron, modulador, generador de potencia de radiofrecuencia, etc. Antes de cada proceso de medidas, se genera un fichero con los ajustes o settings que se emplearán en la medida.

#### Capítulo 4: Interfaz gráfica en MATLAB para adquisición de medidas automáticas del klystron

Es este caso, se realizó un fichero con extensión ‘.m’, que está basado en definir el tipo de medida que se va a realizar, entre las dos posibilidades que tenemos, *measurement\_type* = ‘PinVsPout’ o *measurement\_type* = ‘Frequency’, así que cambiaremos el valor de dicha variable según sea la medida que se va a realizar.

Después lo que se fija es el modelo del klystron, en este caso el *klystron\_model*=‘VKP-8352A’, y también fijamos el número de serie para identificarle correctamente, en este caso es el *klystron\_serial\_number*=‘004’

Se fija la tensión y la corriente del haz en este caso ‘*Vbeam\_KV*=50’ KV e ‘*Ibeam\_A*=21’ Amperios, respectivamente. Después se fijará la frecuencia de operación del klystron, en nuestro caso, ‘*Freq\_MHz* = 352.21’.

Como no es una onda continua lo que vamos a medir, sino pulsada, es necesario fijar la duración de los pulsos tanto de alto voltaje como de radiofrecuencia. Para ello, se configura de la siguiente manera, para alto voltaje ‘*HV\_pulse\_duration\_ms*= 3.4’ y para RF ‘*RF\_pulse\_duration\_ms* = 3.3’ ambos en milisegundos. Otro parámetro que se necesita es la frecuencia de repetición de pulso ‘*Repetition\_frequency\_Hz* = 14’ en Hercios.

En el caso del siguiente ejemplo fichero de configuración, como se realiza una medida de tipo ‘PinVsPout’, se ‘barre’ la potencia del generador de RF, para así obtener la curva de la relación potencia de salida frente a potencia de entrada. Por eso, se define un vector de las potencias que vamos a establecer con el generador de RF, para este ejemplo se decidió ‘*Pgen\_dBm*=[-10:1:0]’ en dBm.

Posteriormente, se configuran tres parámetros importantes por cada uno de los dos detectores de potencia (driver y klystron), y también las pérdidas del cable desde el detector hasta el osciloscopio. Para el detector del driver (fijamos el acoplamiento, la pendiente, y la tensión de salida), igualmente para el detector del klystron. Acoplamiento para el driver ‘*C1\_dB*=60’ dB, pendiente para el detector del driver, ‘*Slope1\_V\_dB*=-0.025’ V/dB y la tensión de salida ‘*Vo1\_V*=1.0792’. Las pérdidas del cable son ‘*Lcable\_dB* = 0.7’, dB. Los parámetros del detector del klystron son: acoplamiento


*'C2\_dB=63.7'* dB, pendiente, *'Slope2\_V\_dB = -0.0244'* V/dB y la tensión de salida *'Vo2\_V=1.0844'* Voltios.

Este fichero de configuración sirve como modelo para los que se puedan crear en un futuro, bastaría con modificar los datos de las variables y guardarlos con otro nombre identificativo del fichero. En el caso del siguiente ejemplo, se guardó con el nombre *'config50kv3\_3ms.m'*. De esa manera, sabemos que la tensión del modulador es 50 kV y que la duración del pulso para la radiofrecuencia es de 3.3 ms. Podemos ver el fichero de configuración con los valores ya mencionados, en el siguiente código MATLAB:

**% Fichero de configuración**

```
measurement_type = 'PinVsPout';% Para el Tipo de medida 'Pin vs. Pout'
%measurement_type = 'Freq Response' %Para medida de respuesta en frecuencia
Klystron_model='VKP-8352A';
Klystron_serial_number='004';
Vbeam_KV = 50;
Ibeam_A = 21;
Freq_MHz = 352.21;
HV_pulse_duration_ms = 3.4;
RF_pulse_duration_ms = 3.3;
Repetition_frequency_Hz = 14;
Pgen_dBm = [-10:1:0];
% power detector (driver)
C1_dB = 60;
Slope1_V_dB = -0.025;
Vo1_V = 1.0792;
Lcable_dB = 0.7;
% power detector (klystron)
C2_dB = 63.7;
Slope2_V_dB = -0.0244;
Vo2_V = 1.0844;
```

## 4.4 Funciones de retrollamada (callback) de los objetos del GUI

Se van a definir ahora las funciones de ‘callback’ del código, para saber realmente qué y cómo ejecutan los comandos, cuando son llamadas mediante la acción sobre los objetos de la GUI. Existen además dos funciones básicas para comunicarnos con la interfaz gráfica son ‘get’ y ‘set’. Para obtener el estado de un ‘checkbox’, éste se refiere a una casilla de activación , si está activada, y queremos saberlo, en el código añadiremos la instrucción: `state=get(handles.nombre_checkbox, 'value')`. Si está activada, `state = '1'`, si no `state = '0'`.

Para representar un dato en un ‘statictext’, de la GUI, se utilizará la siguiente instrucción: `set(handles.nombre_textbox, 'String', num2str(handles.nombrevariable))` utilizamos la función ‘num2str’ para convertir un valor numérico en un ‘string’, ya que en los ‘textbox’ se muestran strings.

### 1) Función de ‘callback’ del objeto: botón ‘CONNECT’

Definición de la función: ***function connect\_Callback(hObject, eventdata, handles)***

Realiza una serie de acciones que se describen a continuación:

- Se cierran instrumentos que se hayan creado previamente, con la función ‘`fclose(instrfind)`’
- Crea el objeto VISA (Virtual Instrument Standard Architecture) que será el que se asigna al osciloscopio, cuyo nombre es ‘`handles.oscilo`’, cuando queramos comunicarnos con el osciloscopio, nos referiremos a la variable creada para el osciloscopio.
- Configura todo lo relacionado con ella, la dirección VISA en el bus GPIB, en nuestro caso:

```
handles.oscilo=visa('tek','GPIB8::1::INSTR');
```

objeto            fabricante    placa    dirección del instrumento

- Configura los 4 canales, fondos de escala, voltios por división, ‘trigger’, activamos los 4 canales, elegimos modo de adquisición, medidas en pantalla (en nuestro caso seleccionamos 4 medidas, 1 por canal), el tipo de medida que queremos, el ancho

de banda. Una vez tenemos el osciloscopio configurado completamente se carga el fichero de configuración de la medida. (Se comenta la funcionalidad, el código está disponible en el ANEXO I).

## 2) Función de ‘callback’ del objeto: botón ‘LOAD CONF.’

Definición de la función: ***function loadconf\_Callback(hObject, eventdata, handles)***

- Esta función realiza la carga del fichero de configuración de la medida, que se explicó, y carga las variables de éste, en variables pertenecientes a la estructura handles.
- Representa los valores de las variables iniciales, cargadas desde el fichero en la interfaz gráfica.
- Crea y acciona un ‘timer’, mediante el cual se hace una llamada a otra función dependiente de las interrupciones generadas por el ‘timer’, la cual toma los valores de los voltajes de las señales conectadas en los cuatro canales, y los representa dinámicamente, a medida que el ‘timer’ crea un evento, cada 0.3 segundos.

## 3) Función de ‘callback’ del objeto: botón ‘CAPTURE’

Definición de la función: ***function capture\_Callback(hObject, eventdata, handles)***

- Detiene el timer.
- Se realiza la adquisición de los valores de los voltajes de cada uno de los canales. Para ello se reinicia en primer lugar el valor del contador del promediado, es decir, una vez tomamos el modo de adquisición ‘average’, necesitamos resetear el contador del promediado, ya que, debido a que realiza una media móvil, podemos arrastrar un cierto error en cada medida.
- Representa cada nueva adquisición en la interfaz de la GUI.
- Dibuja en los ejes de la gráfica, cada nuevo punto adquirido, junto con los anteriores.
- Va adquiriendo en una matriz el resultado de las variables que nos interesen luego exportar al archivo .csv
- Reinicia el timer.

#### 4) Función de ‘callback’ del objeto: botón ‘COMPLETE’

Definición de la función: ***function complete\_Callback(hObject, eventdata, handles)***

- Realiza una captura del entorno GUI con los resultados y lo exporta en un archivo .jpg, que se guarda en la misma ubicación donde se encuentra el GUI.
- Exporta la matriz de resultados en un archivo .csv que lo podemos guardar con el nombre que interese, y lo podemos guardar donde deseemos. Posteriormente, puede abrirse con cualquier programa que interprete este tipo de documentos, por ejemplo, en EXCEL.

#### 5) Función de ‘callback’ del objeto: botón ‘PREVIOUS’

Definición de la función: ***function previous\_Callback(hObject, eventdata, handles)***

- Varía la posición en el vector de potencias de RF, decrementando en una unidad el contador ‘handles.i’, para situarnos en la potencia anterior, que se va a configurar del vector de potencias del generador, en ese momento, (en el caso de medida Pin/Pout).
- Cambia la posición decrementando en uno el contador ‘handles.i’, para cambiar a una frecuencia anterior, en el vector de frecuencias, en el caso de estar realizando la medida frecuencial.

#### 6) Función de ‘callback’ del objeto: botón ‘NEXT’

Definición de la función: ***function next\_Callback(hObject, eventdata, handles)***

- Varía la posición en el vector de potencias de RF, incrementando en una unidad el contador ‘handles.i’, para situarnos en la potencia que se va a configurar, en ese momento, (en el caso de medida Pin/Pout).
- Cambia la posición incrementando en uno el contador ‘handles.i’, para cambiar la frecuencia en el vector de frecuencias, en el caso de estar realizando la medida frecuencial.

### 7) Función de ‘callback’ del objeto: botón ‘Ibeam\_button’

Definición de la función: ***function Ibeam\_button\_Callback(hObject, eventdata, handles)***

- Comprueba el estado del ‘checkbox’, correspondiente a la señal del canal 2, cuando conectamos ‘Ibeam’ a dicho canal.
- Si está activado, cambia la cabecera del fichero .csv, poniendo ‘Ibeam’ en la posición segunda comenzando por la izquierda, de los títulos de los títulos de la tabla de resultados.

### 8) Función de ‘callback’ del objeto: botón ‘Pklyrv\_button’

Definición de la función: ***function Pklyrv\_button\_Callback(hObject, eventdata, handles)***

- Comprueba el estado del ‘checkbox’, correspondiente a la señal del canal 2, cuando conectamos ‘Pklyrv’ a dicho canal.
- Si está activado, cambia la cabecera del fichero .csv, poniendo ‘Pklyrv’ en la posición segunda comenzando por la izquierda, de los títulos de la tabla de resultados.

### 9) Función de ‘callback’: timer

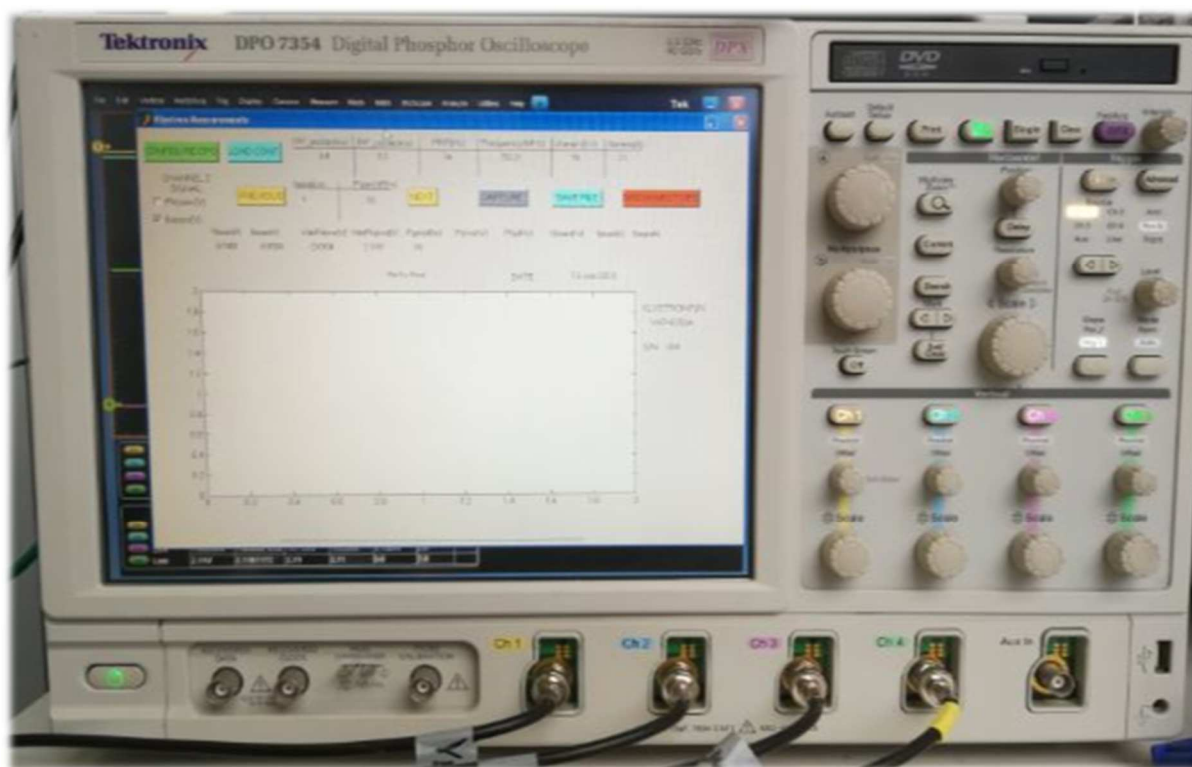
Definición de la función: ***function timerCallback(hObject, eventdata, handles)***

- Realiza las medidas de los cuatro canales y las va representando en el GUI, a medida que hay un evento de timer, va actualizando dinámicamente el valor actual, facilita la percepción de los cambios en los valores.



#### Capítulo 4: Interfaz gráfica en MATLAB para adquisición de medidas automáticas del klystron

En la siguiente figura se muestra el programa de MATLAB para la automatización de medidas del klystron incorporado en el osciloscopio Tektronix DPO 7354 de ESS Bilbao.



**Figura 4.11** Software de medidas automáticas en el osciloscopio de ESS Bilbao

## Capítulo 5: Medida de alta potencia por calorimetría

En el proceso de reacondicionamiento de los klystron dentro del RF test stand es necesario realizar medidas lo más exactas posibles de la potencia de señal de radiofrecuencia que se obtiene en saturación a la salida de estos amplificadores. Esta medida es muy importante, debido a que las cavidades aceleradoras requieren una potencia precisa para desarrollar el campo eléctrico acelerante especificado. Una potencia más baja puede degradar la aceleración de las partículas y una potencia demasiado alta puede dañar las cavidades u otros sistemas. Debido a esa premisa se realizan diferentes métodos de medición, de manera que se obtengan valores fiables y contrastados.

### 5.1 Técnicas de calorimetría para medida de alta potencia de RF

Dentro de los diversos métodos empleados para medida de potencia de radiofrecuencia, podemos encontrar medidores de potencia basados en diodos detectores, termopares, termistores, medidores bolométricos, etc. Habitualmente estos métodos son adecuados para niveles bajos y medios de potencia. En cambio, cuando se trata de medir altas potencias, en algunos casos se recurre a técnicas de calorimetría.

La calorimetría [12] permite, en general, estimar el calor o la energía necesaria para aumentar la temperatura de una determinada cantidad de un material caracterizado por su calor específico, de acuerdo con la siguiente expresión:

$$Q(J) = c_p \left( \frac{J}{kg \cdot ^\circ C} \right) \cdot m(kg) \cdot \Delta T(^{\circ}C) \quad (5.1)$$

Donde:

$Q$ : calor necesario (Julios, J)

$c_p$ : calor específico del material ( $\frac{J}{kg \cdot ^\circ C}$ )

$m$ : masa del material (kg)

$\Delta T$ : Incremento de temperatura del material ( $^{\circ}C$ )

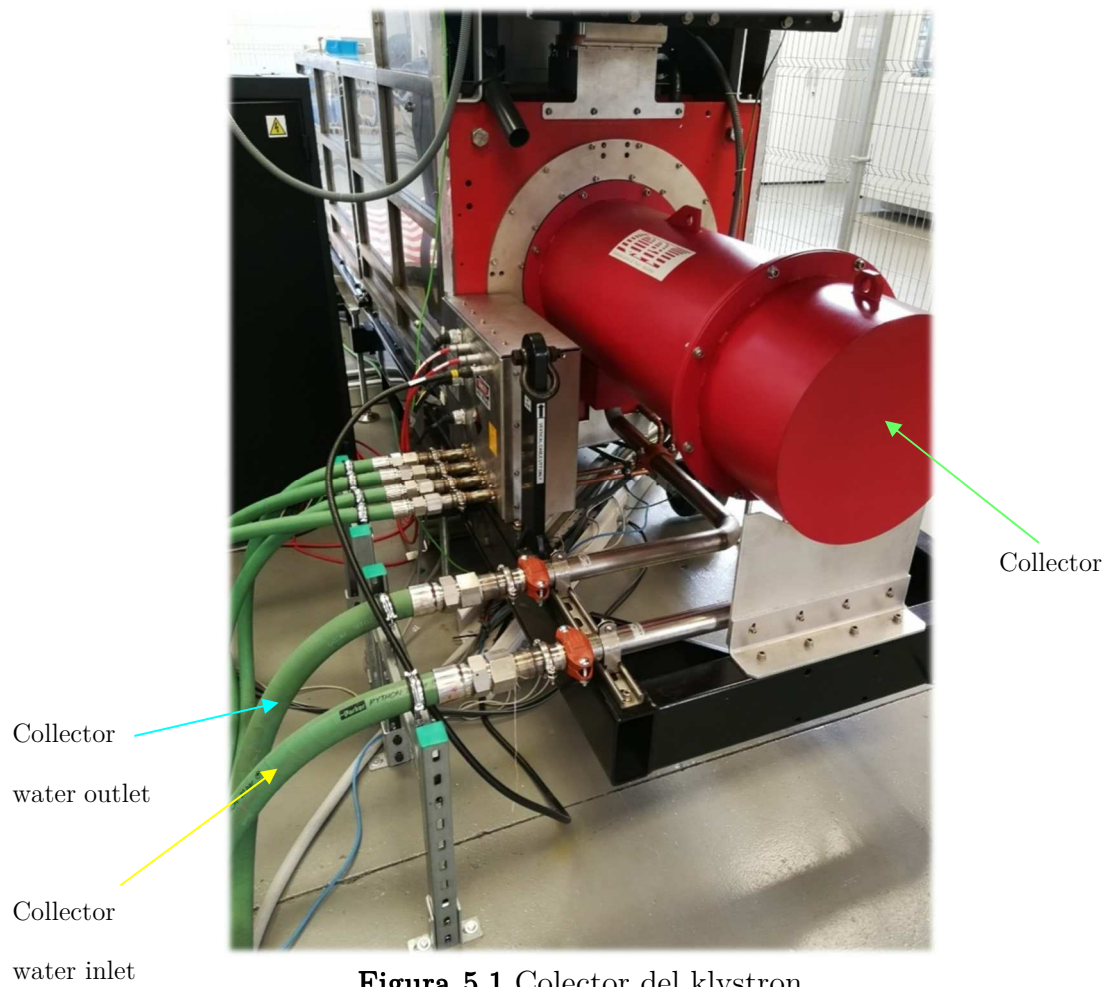
Las técnicas de calorimetría se pueden aplicar a la medida de potencia de RF, ya que se reúnen las condiciones necesarias y se dispone de los valores requeridos. Esto conlleva conocer los parámetros que intervienen en el proceso.

La instalación donde se lleva a cabo este procedimiento dispone de un sistema de amplificación de señal de radiofrecuencia basado en klystron, el cual está alimentado con una fuente de alta tensión pulsada (modulador) y un preamplificador driver (al cual está conectado un generador de señal de RF). Posteriormente la salida de RF del klystron se conecta a una carga de RF, por medio de guías de onda.

El klystron está enfriado por un sistema de refrigeración líquido que usa agua desionizada. En particular, el colector está conectado a dos tuberías de entrada y salida. La primera introduce el agua y la segunda lo extrae para llevarlo al intercambiador de calor. El agua caliente va a un enfriador de agua o ‘water chiller’, del chiller vuelve enfriada otra vez al depósito. El interior del colector presenta unos surcos en la superficie, que permite la irrigación del agua llegando a todos los puntos de la superficie interna y realizando una refrigeración apropiada. Se puede observar en la siguiente imagen la conexión al colector.

Es importante la adecuación del sistema de refrigeración, que lleva un especial mantenimiento periódico, y requiere mantener un sistema de control del circuito de refrigeración de agua, independiente al control del resto de los elementos de potencia del RF test stand. Por ello, se realizan pruebas separadas, de niveles de caudal, presión, contenido en PH, cloración, desionización, desoxigenación, entre otras. Se pretende garantizar siempre con todo ello, una buena calidad del sistema, evitando que los elementos de la cadena de amplificación se puedan sobrecalentar, y así exceder los límites de seguridad establecidos en las recomendaciones de los fabricantes.

El circuito de refrigeración consta de los siguientes elementos principales: un depósito o vaso de expansión donde se almacena el agua, una bomba, filtros de desmineralización de agua, sensores de medida. El depósito de agua se rellena con nitrógeno para evitar la oxigenación adicional. Los circuitos de salida y retorno de agua están controlados individualmente por llaves de paso, sensor de temperatura, medidores de presión, caudalímetros, y una electroválvula de protección contra sobrepresiones.



**Figura 5.1** Colector del klystron



**Figura 5.2** Llave de paso y manómetro



**Figura 5.3** Intercambiador de calor



**Figura 5.4** Vaso de expansión y medidores de calidad del agua (pH, O<sub>2</sub>, conductividad eléctrica [ $\mu\text{S}/\text{cm}$ ], temperatura [ $^{\circ}\text{C}$ ])

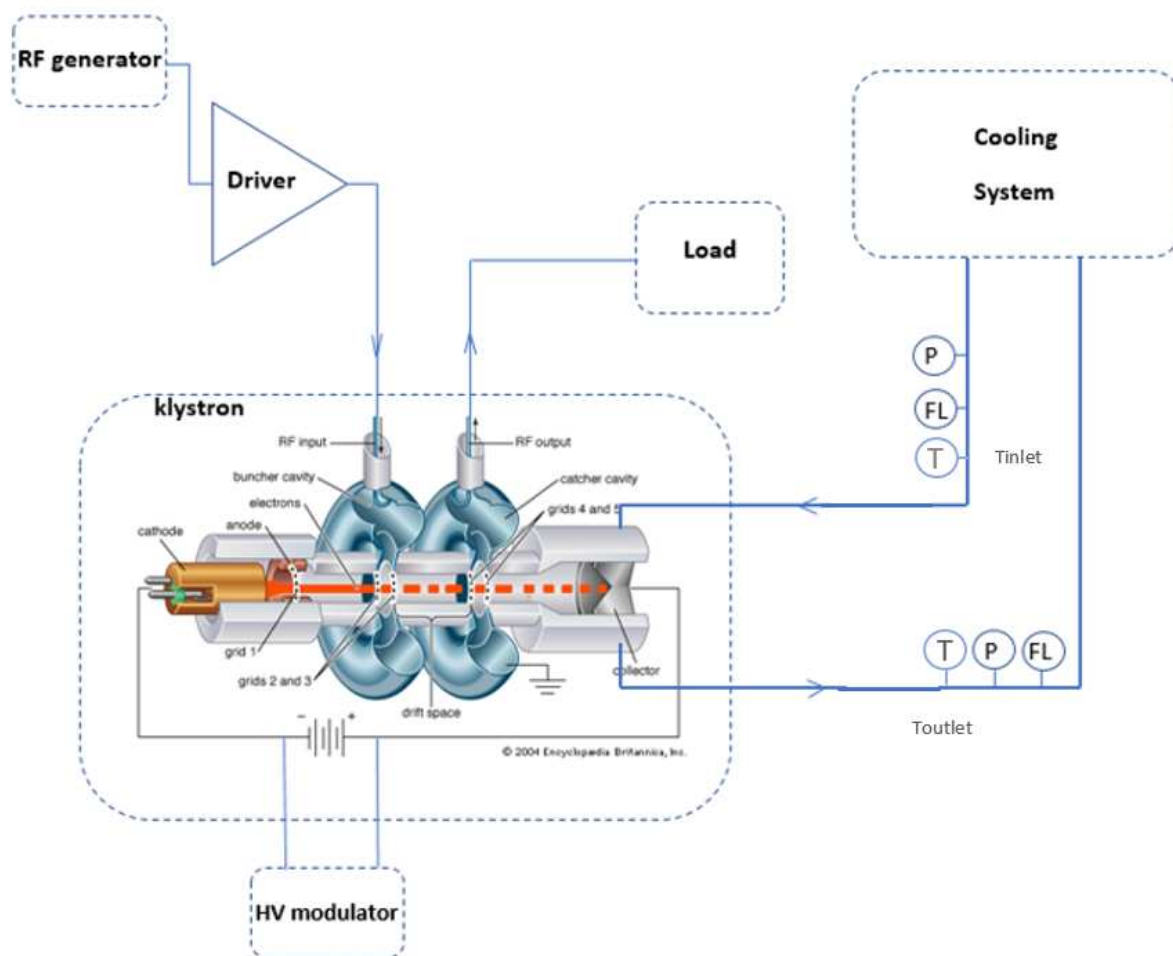


**Figura 5.5** Distribuidor de circuitos de agua independientes, compuestos por dos tubos cada uno (de ida y de retorno)

Todo el proceso está controlado por el sistema de control de agua, y éste a su vez está comunicado con el sistema de interlocks. Cualquier aumento excesivo de temperatura del agua, aumento de la presión, o disminución del caudal (por posibles fugas), está protegido debidamente, con una señal de ‘Agua OK’ que se recibe en la ‘control room’ (si se encuentra en buenas condiciones) o se recibirá una alarma de que ha ocurrido alguna anomalía en el circuito del agua. A partir de la cual se cortará el suministro de alta tensión y otros factores eléctricos para evitar sobrecargas y fallos de refrigeración, que se puedan traducir en averías relevantes de componentes.

Como se observará en el siguiente esquema, se utilizan sensores de temperatura PT100, para temperatura a la entrada (Tinlet), y a la salida (Toutlet), los cuales nos ofrecerán una medida bastante exacta. Después se toma la medida de temperatura en la tubería de retorno del agua una vez haya circulado por el interior del colector del klystron. Tal y como se muestra en la siguiente figura.





**Figura 5.6** Esquema de las sondas de medida temperatura, presión y caudal del circuito de agua.



**Figura 5.7** Sensores de temperatura PT100, y de presión



**Figura 5.8** Caudalímetro

Aprovechando que se conocen y se monitorizan los valores de temperatura y caudal del refrigerante, se puede medir la potencia disipada en el colector, como se ha explicado anteriormente.

A través de la siguiente expresión (5.2) se puede obtener el calor necesario por unidad de tiempo [13], cuyo valor es equivalente a la potencia que se disipa en el colector, que se mide en vatios ( $W=J/s$ ). Dicha expresión deriva de (5.1), donde se ha obtenido la energía disipada por unidad de tiempo (esto es, la potencia disipada), y la masa por unidad de tiempo se ha sustituido por el producto del caudal por la densidad del agua.

$$P_{dis}(W) = c_p(J/kg \cdot ^\circ C) \cdot \frac{Fl(lpm)}{60} \cdot \rho(kg/l) \cdot \Delta T(^{\circ}C) \quad (5.2)$$

Donde:

$P_{dis}$ : Potencia disipada en el colector ( $W=J/s$ )

$c_p$  : Calor específico a presión constante del refrigerante ( $4180 \frac{J}{kg \cdot ^\circ C}$  para el agua)

$Fl$ : Caudal de refrigerante (litros por minuto, lpm)

$\rho$  : Densidad de masa del refrigerante (aproximadamente 1 kg/l para el agua)

$\Delta T$  : Incremento de temperatura del líquido refrigerante ( $^{\circ}C$ )



A partir de determinar las dos temperaturas ( $T_{inlet}$  y  $T_{outlet}$ ) conocemos el valor de  $\Delta T$  que nos permitirá calcular la potencia disipada en el colector.

Por otro lado, la potencia eléctrica disipada en el colector se puede calcular con la siguiente expresión:

$$P_{dishV}(W) = V_{beam}(V) \cdot I_{beam}(A) \cdot DC_{HV} \quad (5.3)$$

Donde:

$V_{beam}$  : Tensión de haz en el klystron

$I_{beam}$  : Corriente de haz del klystron

$DC_{HV}$  : Ciclo de trabajo (Duty Cycle) de alto voltaje. Este valor es igual al tiempo de duración del pulso por la frecuencia de repetición del pulso.

Como se ha explicado anteriormente, el hecho de calcular la potencia del klystron mediante calorimetría, nos permite fundamentalmente, saber la potencia de señal de RF que sale amplificada del klystron. Para saber este dato, se realiza el siguiente procedimiento:

1º Alimentando el klystron con alto voltaje, y sin potencia de RF se mide la potencia que se disipa en el colector. Para determinar una medida de ésta, es necesario esperar un tiempo a que la temperatura del agua en el colector se estabilice.

En primer lugar, hay que determinar la diferencia de temperaturas del refrigerante a la entrada y salida del colector sin RF y con alta tensión:

$$\Delta T_{HV} = T_{outlet} - T_{inlet} \quad (5.4)$$

Para dicho valor de  $\Delta T_{HV}$  se obtiene la potencia disipada en el colector en vatios.

$$P_{dishV} = c_p \left( \frac{J}{kg \cdot ^\circ C} \right) \cdot \frac{Fl(lpm)}{60} \cdot \rho(kg/l) \cdot \Delta T_{HV} \text{ (} ^\circ C \text{)} \quad (5.5)$$

2º Después de eso, se activará el generador de RF, operamos el klystron en saturación con la señal de RF adecuada, y nuevamente se dejará un tiempo de margen para que la temperatura del colector alcance un valor estable y se toma la medida:

$$\Delta T_{HV\_RF} = T_{outlet} - T_{inlet} \quad (5.6)$$

Y nuevamente, se calcula la potencia que se disipa en el colector:

$$P_{disHV\_RF} = c_p \left( \frac{J}{kg \cdot ^\circ C} \right) \cdot \frac{Fl(lpm)}{60} \cdot \rho(kg/l) \cdot \Delta T_{HV\_RF} (^\circ C) \quad (5.7)$$

Una vez se hayan adquirido las medidas de potencia con el modulador activado y generador de RF desactivado, y la medida con el modulador activado y el generador de RF activado, se calcula la potencia de radiofrecuencia que sale del klystron ya amplificada.

Por tanto, la potencia de RF es la diferencia entre ambas potencias ( $P_{disHV} - P_{disHV\_RF}$ ). Según el principio de conservación de la energía, la potencia que se deja de disipar en el colector, en condición de RF activada, no se pierde si no que se transforma en potencia de señal de RF que se disipa en la carga.

Teniendo en cuenta el concepto anterior, se calcula la potencia promedio de RF que sale del klystron como:

$$P_{RFavg} = P_{disHV} - P_{disHV\_RF} \quad (5.8)$$

Con todo ello tenemos la potencia promedio de RF disponible a la salida del klystron, no obstante, se mide la potencia de pico de RF de la siguiente manera:

$$P_{RFpk} = \frac{P_{RFavg}}{DC_{RF}} \quad (5.9)$$

## 5.2 Interfaz gráfica de usuario para medidas de potencia con calorimetría con el software LabView

Para medir la potencia promedio de RF mediante calorimetría en la instalación de pruebas de RF de alta potencia de ESS Bilbao, se realizó una interfaz gráfica de usuario. Esta interfaz facilita la adquisición de las medidas de los sensores y su calibración, así como la introducción de los parámetros requeridos para los cálculos de calorimetría, y así poder obtener unas medidas de potencia fiables. Todo ello posibilita obtener los resultados de la medida de una manera gráfica y amigable.

A modo de introducción y para explicar el programa en LabView® de National Instruments, se realiza una aclaración sobre en qué consiste fundamentalmente este software y qué partes tiene. Todo programa en LabView dispone de un panel frontal (front panel), en el que el usuario puede añadir a su gusto los objetos que le permitirán realizar medidas con una cierta abstracción del código, es decir, manipulando objetos como botones, interruptores, potenciómetros, etc. Así el usuario, es capaz de utilizar el programa y a su vez, abstraerse de la dificultad del código, centrándose en el proceso de medida/cálculo en sí mismo. Se puede observar a continuación un ejemplo de front panel:

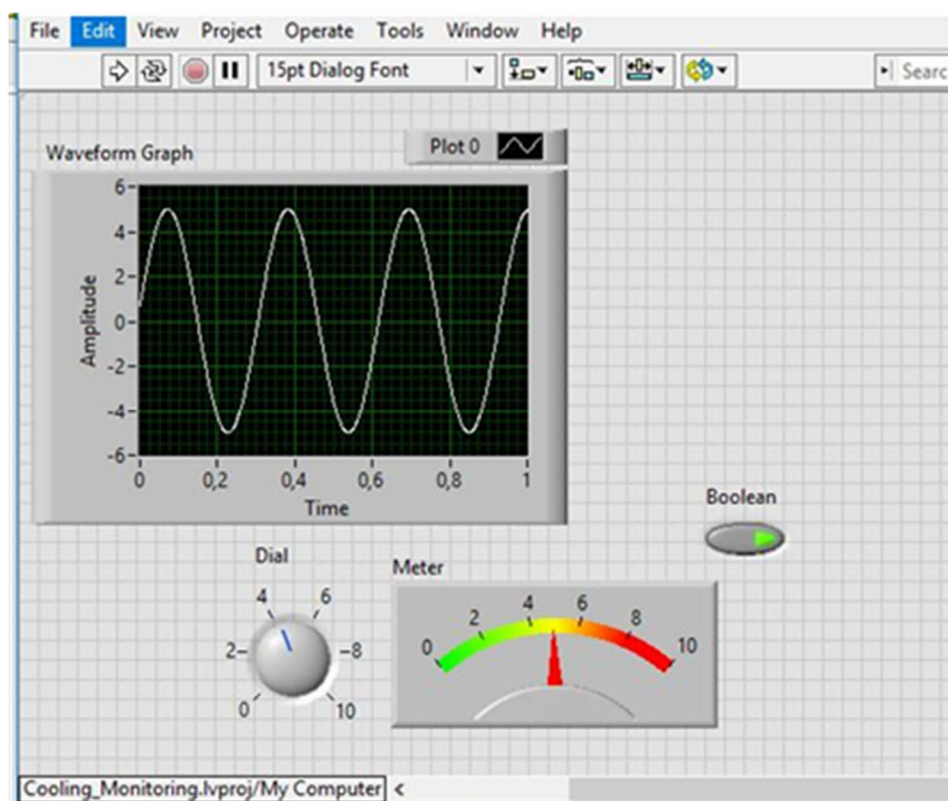
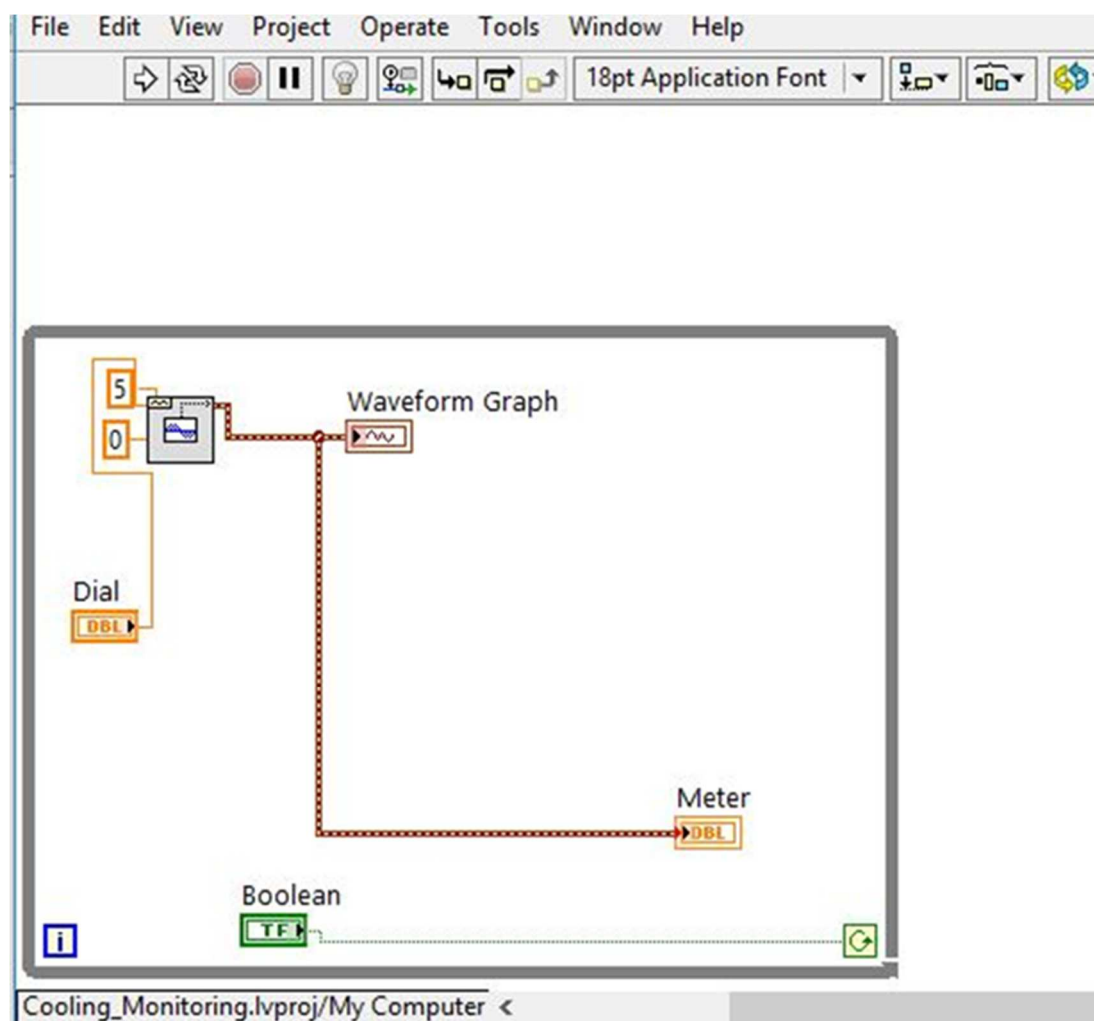


Figura 5.9 Ejemplo de panel frontal

La funcionalidad que sustenta el panel frontal se describe en el diagrama de bloques de LabView (block diagram). En él, se configuran las conexiones entre los elementos, se añaden variables, estructuras, etc. De esa manera se programan los elementos para que se realicen las operaciones deseadas. Siguiendo con el ejemplo anterior de panel frontal, se describirá su funcionamiento a partir de su diagrama de bloques:



**Figura 5.10** Diagrama de bloques del panel frontal del ejemplo

Observamos en el diagrama de bloques anterior la función que realiza nuestro VI (Virtual Instrument). En este caso se observa una estructura ‘while’, la cual se va ejecutando, mientras el interruptor booleano, esté pulsado, en el momento que le desactivemos dejan de ejecutarse las acciones de su interior. Estas acciones corresponden a un generador de señal sinusoidal, que genera un seno de amplitud cinco, con fase cero, y la frecuencia se va variando, con el ‘Dial’, se visualiza conectado a un visualizador y a un medidor.

Se desarrolló una interfaz de usuario ad-hoc particularizada para ESS Bilbao, que permite realizar medidas de potencia de RF mediante el método de calorimetría. Dicho programa se utiliza para las medidas de alta potencia en los reacondicionamientos de los amplificadores klystron. Se diseñó el siguiente panel frontal.

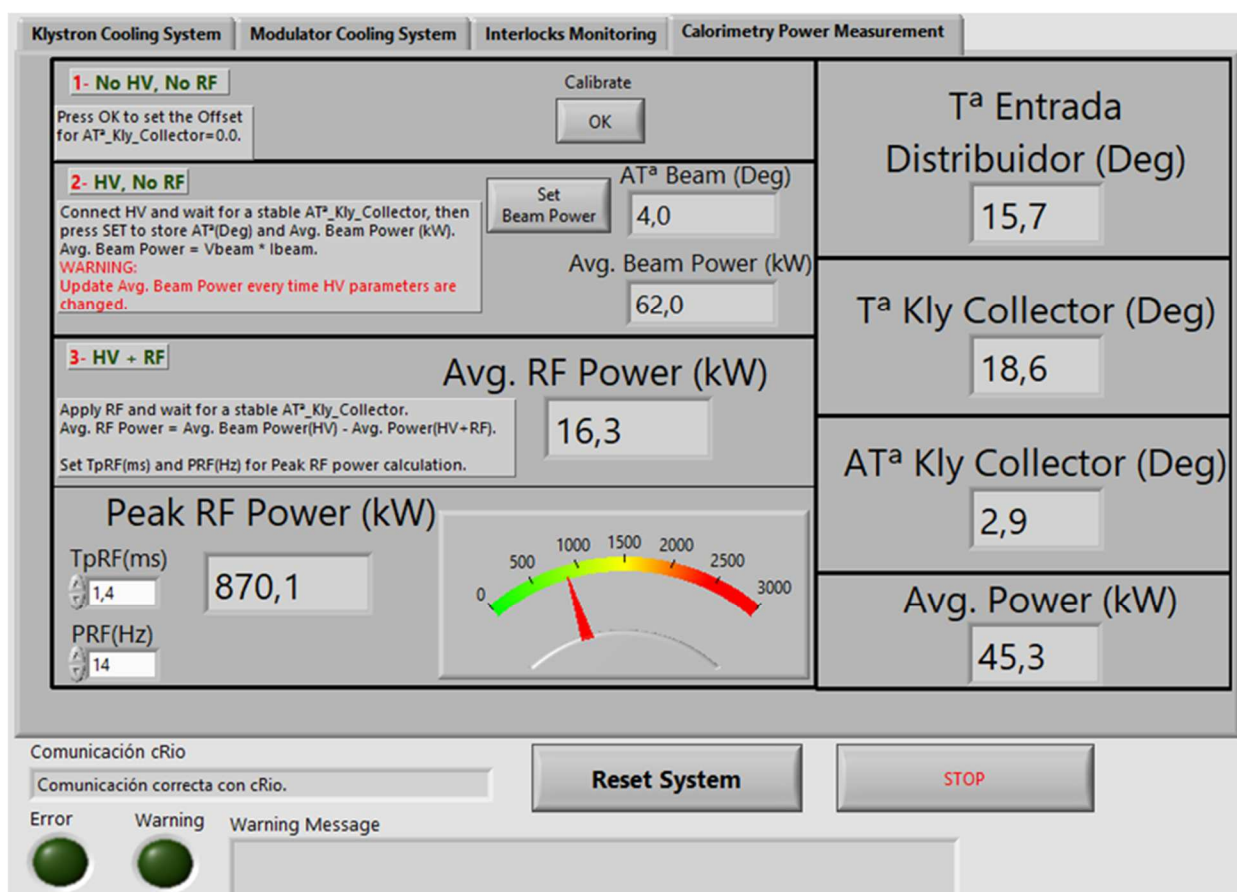


Figura 5.11 Panel frontal del programa para la medida de alta potencia mediante calorimetría

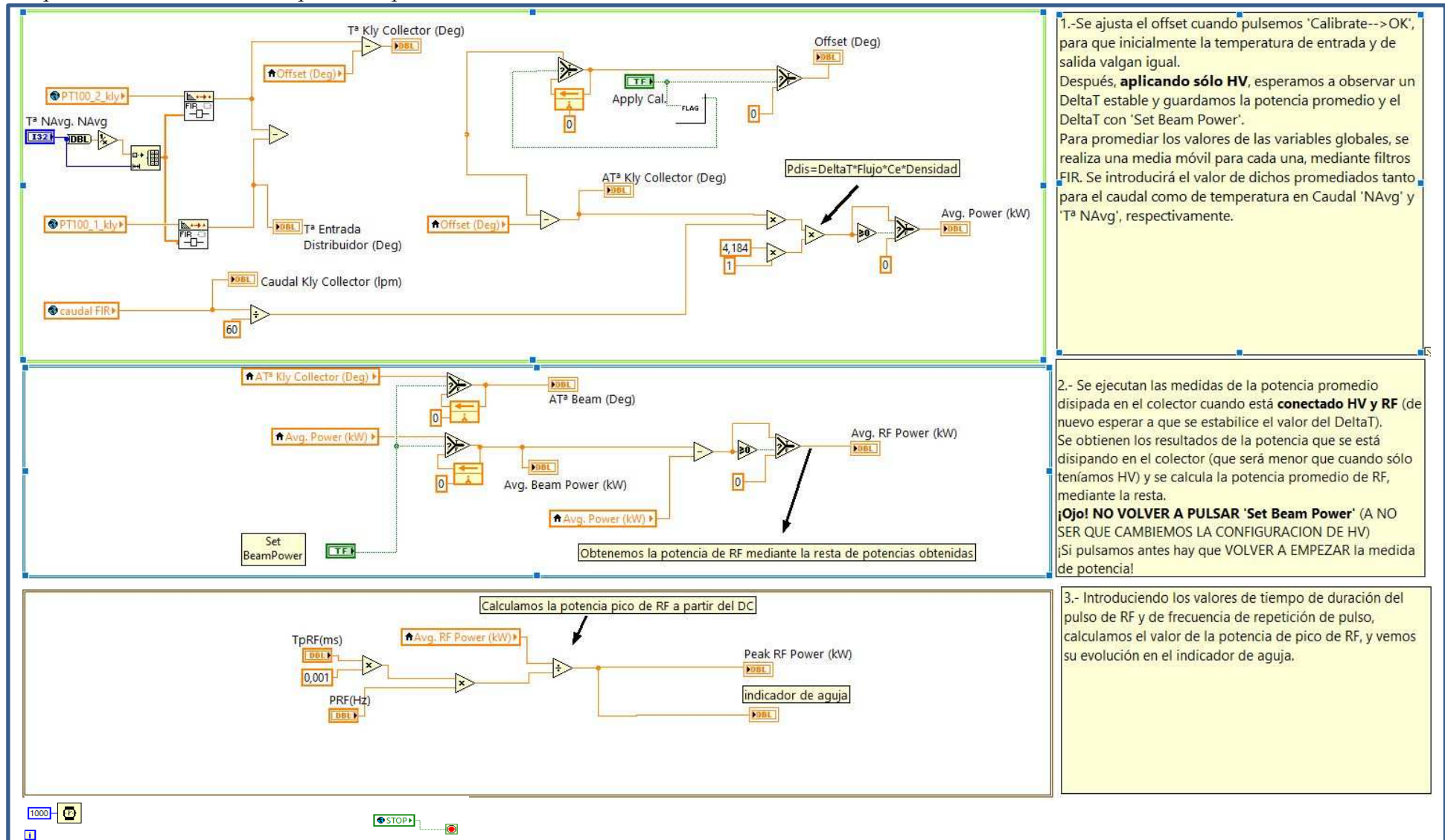


Figura 5.12 Diagrama de bloques del programa desarrollado de medida de potencia

En el diagrama de bloques anterior se muestran las tres partes principales mencionadas anteriormente, todas ellas englobadas en una estructura ‘while’ (recuadro azul), que se va ejecutando con un periodo de un segundo, hasta que se interrumpa su ejecución a través del ‘STOP’ global del programa principal, ‘main.vi’. Por tanto, seremos capaces de conocer la evolución de la potencia tanto disipada en el colector, como potencia promedio de salida de RF del klystron.

En los siguientes puntos se describen los pasos que hay que seguir, para realizar una medida de alta potencia mediante la interfaz de usuario diseñada, que se presentó previamente en la Figura 5.11.

**1º)** Sin aplicar alto voltaje y sin aplicar potencia de RF, ajustaremos un offset para que la referencia de temperaturas cuando no aplicamos señal sea la misma. Ello se debe a que los termopares no miden de igual manera. El offset se establece cuando se pulsa ‘Calibrate offset’. Una vez hayamos fijado el offset se observará un ‘ $\Delta T$ ’ inicial de valor cero.

**2º)** Una vez conectado el alto voltaje, sin conectar la potencia de RF, se espera a que se adquiera una temperatura estable, traducido en un  $\Delta T$  estable, para fijar dicho valor de  $\Delta T$ . Una vez fijado el valor del  $\Delta T$  pulsaremos el botón ‘Set beam power’, el cual además fija la ‘Avg. Beam Power (kW)’. En el indicador de Avg. Beam Power (kW), se visualizará el valor de la potencia promedio de haz.

**3º)** Cuando está conectada la señal del generador de RF, se procede a calcular el valor de la potencia de radiofrecuencia. Para ello, debido a que estamos trabajando con señales pulsadas, es necesario conocer el ciclo de trabajo. Se necesitan, por tanto, dos valores que son la duración del pulso de RF,  $T_{pRF}(\text{ms})$ , medida en milisegundos, la frecuencia de repetición de pulsos,  $PRF(\text{Hz})$ , medida en Hz. Ambas respectivamente mostradas en la interfaz de usuario.

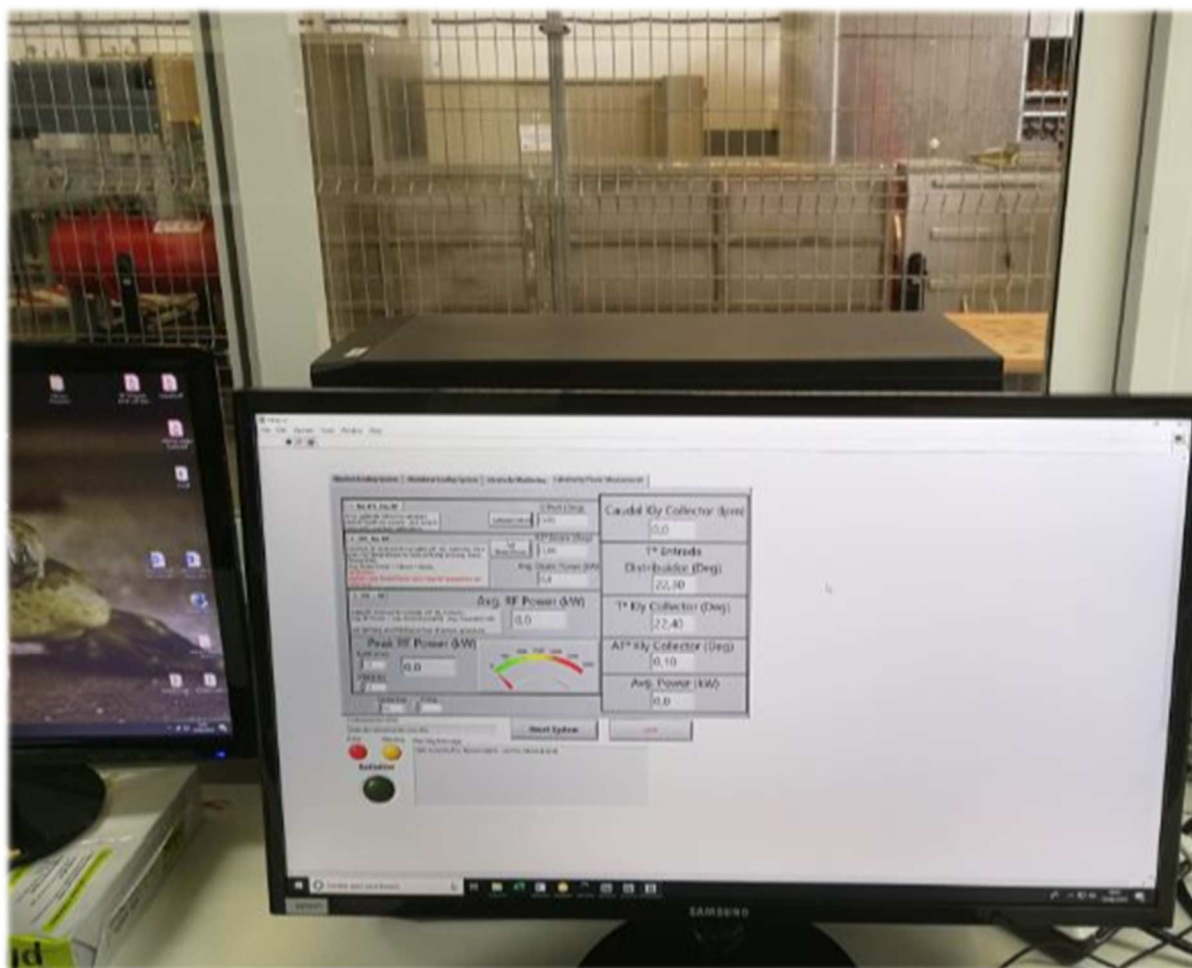
**4º)** Dejamos conectado (como se ha comentado antes) el alto voltaje y la señal RF, mientras va estabilizándose la temperatura, por ende, el  $\Delta T$ , se calcula la potencia promedio disipada en el colector cuando tenemos señal de RF. Como ahora se disipa menos potencia en el colector, a la potencia promedio medida solamente con alto voltaje, se le resta la potencia actual y obtenemos la potencia promedio de RF que hay a la salida del klystron, RF Avg. power(kW). Además, con el cálculo realizado previamente del ciclo



de trabajo se presenta el valor de la potencia pico de RF ,RF peak power, mostrada en el medidor y numéricamente.

5º) Introduciendo los valores de la duración del pulso de RF,  $T_{pRF}(\text{ms})$ , y de la frecuencia de repetición de pulsos,  $PRF(\text{Hz})$ , se calcula el ciclo de trabajo de RF como el producto de ambos. A partir de él se obtiene el valor de la potencia de pico de RF como, la potencia promedio de RF entre el valor del Duty Cycle de RF.

A continuación, se presenta una imagen del software de calorimetría implementado en la sala de control del RF Test Stand de ESS Bilbao, donde se llevaron a cabo las medidas.



**Figura 5.13.** Software de calorimetría implementado en la sala de control del RF Test Stand



### 5.3 Cálculos de la medida de potencia de RF por calorimetría

En la Figura 5.11, se muestra un caso de una medida de potencia de RF usando calorimetría para el klystron VKP-8352A de CPI, Inc., S/N 002, operado con los siguientes parámetros del modulador:  $V_{\text{beam}} = 85 \text{ kV}$ ,  $I_{\text{beam}} = 32.7 \text{ Amp}$ ,  $t_{\text{pulseHV}} = 1,6 \text{ msec}$ ,  $\text{PRF} = 14 \text{ Hz}$ . El ciclo de trabajo correspondiente era de  $\text{DCHV} = 2,24\%$  y la potencia eléctrica esperada (según la ecuación 5.3) era de  $62,3 \text{ kW}$ .

Por su parte, la señal de RF era asimismo pulsada, con  $t_{\text{pulseRF}} = 1.4 \text{ msec}$ ,  $\text{PRF} = 14 \text{ Hz}$ . El ciclo de trabajo,  $\text{DCRF} = 1,96\%$ ).

El caudal del circuito de refrigeración del colector del klystron era de 222 litros por minuto. Se asumía, asimismo, y considerando el calor específico del agua de  $4180 \text{ J/kg.K}$ , su densidad de  $1 \text{ kg/dm}^3$ .

**Paso 1:** Se calibró el offset de los termopares para que  $T_{\text{outlet}} = T_{\text{inlet}}$  (esto es,  $T^{\text{a}} \text{ Kly Collector} = T^{\text{a}} \text{ Entrada Distribuidor}$ ), en este caso  $15.7$  grados centígrados.

**Paso 2:** Se alimentó el klystron con alta tensión, sin RF. Tras esperar a que se estabilizara la temperatura de salida del colector del klystron unos 2 minutos, se obtuvo un incremento de temperatura provocado por el haz de electrones,  $\Delta T_{\text{Beam}} = 4.0$  grados, que se correspondía con una potencia promedio disipada de  **$62.0 \text{ kW}$**  (usando la ecuación 5.5). Este valor era muy cercano al esperado de  $62.3 \text{ kW}$ .

**Paso 3:** Se alimentó el klystron con alta tensión y se aplicó RF hasta llegar a saturación. Una vez estabilizada la temperatura de salida del colector del klystron, se obtuvo un incremento de temperatura  $\Delta T = 2.9$  grados, correspondiente a una potencia promedio disipada en el colector de  **$45.3 \text{ kW}$** .

La potencia promedio de RF era la diferencia entre los valores obtenidos en el paso 2 y el paso 3, esto es  **$16.3 \text{ kW}$**  (ecuación 5.8). Teniendo en cuenta el ciclo de trabajo del pulso de RF, la potencia de pico de RF resultó  **$870.1 \text{ kW}$** .

## 5.4 Causas y fuentes de error en la medida

En todo proceso de medida, existen factores externos e internos que impiden el cálculo exacto de los valores de los parámetros que se miden. Las principales fuentes de error que podemos encontrar dentro de nuestro laboratorio ‘RF test stand’ en la medida de potencia de radiofrecuencia del amplificador klystron mediante detectores de potencia son:

- **Error de linealidad en la respuesta de los detectores** de potencia de RF. Los detectores de Minicircuits utilizados tienen una relación aproximadamente lineal entre la potencia de entrada en dBm y la tensión detectada en voltios (con una pendiente de  $-0.025 \text{ V/dB}$ ). Aunque la respuesta de cada unidad se mide previamente y se aproxima por una recta, existe un cierto error, sobre todo al acercarnos a los extremos inferior y superior de su rango dinámico. Este error puede estar alrededor de 0.1 ó 0.2 dB.
- **Imprecisión de los detectores de potencia.** Un pequeño error en la medida de la tensión de salida del detector, por ejemplo 5 mV, se traduce en 0.2 dB de error en la potencia de RF.
- **Error por el acoplo de los acopladores direccionales.** Para medir niveles de potencia de hasta 3 MW de pico (94.77 dBm), se obtienen muestras por medio de acopladores direccionales (de 50 dB de acoplo) seguido de atenuadores (de 30 dB), a fin de acomodar la potencia del klystron al rango dinámico de los detectores de potencia. No es fácil medir con exactitud el acoplo de un acoplador direccional con entrada y salida en guía de onda WR2300 y muestras en coaxial. Típicamente se puede cometer un error cercano a 0.2 dB.
- **Error frecuencial de los detectores.** Los sensores de potencia disponen de una curva de respuesta frecuencial, lo cual genera que el dispositivo no se comporte igual en toda su banda de trabajo. Esa variación en la respuesta frecuencial del sensor puede causar un pequeño error en la medida, que se minimiza calibrando los detectores de potencia a la frecuencia de operación de 352.21 MHz.

- **Relación de Onda Estacionaria (ROE) y directividad de los acopladores direccionales.** El aislamiento finito (y por tanto directividad no ideal) de los acopladores direccionales pueden resultar en errores de medida de potencia en presencia de ondas reflejadas no despreciables, producidas por discontinuidades en las guías de onda. La muestra de la potencia incidente en el puerto acoplado se ve perturbada por la captura de una fracción de la potencia reflejada, la cual se suma vectorialmente a la señal deseada.
- **Estabilidad medioambiental.** Otro factor a tener en cuenta en cuanto a error en la medida es el ambiental. El cambio en condiciones de temperatura, presión, humedad, etc. modifica el comportamiento de todo el sistema, afectando directamente en la medida.

En la siguiente tabla se muestra un resumen del impacto de un pequeño error de décimas de dB en la medida de potencia mediante detectores, traducido a error absoluto de potencia de RF en W o en kW, para un nivel de potencia nominal de 10 W y de 3 MW, respectivamente.

**Tabla 5.1** Valores de error en la potencia de RF debidas al detector

Error (dB)	Error relativo (%)	Error absoluto para 10 W	Error absoluto para 3 MW
0.1 dB	2.5 %	0.25 W	75 kW
0.2 dB	5 %	0.5 W	150 kW
0.3 dB	7.5 %	0.75 W	225 kW
0.4 dB	10 %	1 W	300 kW

Como se puede apreciar, el error absoluto cometido en la medida de potencia de RF con detectores es proporcional al nivel de potencia nominal, resultando inaceptable para los niveles del klystron. No es fácil reducir este error y así mejorar la exactitud de la medida.

En cambio, la medida de alta potencia de RF por el método de calorimetría (aún cuando puede estar afectada por la imprecisión de los sensores de temperatura y de

caudal), contrastada con la potencia eléctrica que proporciona el modulador, arroja resultados con un error estimado por debajo de los 60 kW para la potencia de saturación del klystron (correspondiente a un error relativo de alrededor del 2 %). La exactitud del método de calorimetría mejora cuanto mayores sean las potencias promedio disipadas.

Por ello, habitualmente se emplea el método de calorimetría para obtener el nivel de potencia absoluto del klystron en saturación y, en esas condiciones, calibrar los detectores de potencia. Luego, las medidas de caracterización del klystron en frecuencia y frente a la potencia de entrada se realizan de forma muy rápida con los detectores de potencia, usando el software de medidas automáticas desarrollado en MATLAB para el osciloscopio (descrito en el capítulo 4).

## **Capítulo 6: Conclusiones y líneas futuras**

En este trabajo fin de Máster se ha descrito en profundidad la participación que se llevó a cabo con el Departamento de Ingeniería de Comunicaciones de la Universidad de Cantabria en colaboración con el Consorcio ESS Bilbao en el proyecto de reacondicionamiento de dos amplificadores klystron de alta potencia para ESS ERIC. Este paquete de trabajo forma parte de la contribución en especie de ESS Bilbao a la fuente ESS ERIC, relativa a los sistemas de radiofrecuencia de alta potencia.

En el TFM se han realizado diferentes aportaciones. En primer lugar, se introdujeron dos mejoras en el sistema de enclavamientos rápidos (fast interlocks), encargado de la protección de los sistemas de RF. En segundo lugar, se ha desarrollado software de medida automatizada para la caracterización de los amplificadores klystron.

En cuanto al sistema de enclavamientos rápidos, por un lado, se verificó y se comprobó la viabilidad de realizar un comparador de umbral negativo para señales provenientes del modulador mediante unos cambios en algunos componentes en el hardware de dicho comparador.

Por otro lado, se introdujo una mejora en el comparador de potencia incidente y reflejada de RF en el Klystron, que consistió en modificar el esquema inicial, basado en un amplificador de instrumentación con un slew rate insuficiente. Para ello, hubo que investigar sobre la existencia de un amplificador de instrumentación con mejores prestaciones. El circuito comparador original provocaba una falsa alarma al inicio de cada pulso de RF, antes de que la señal de salida del amplificador de instrumentación se estabilizase. La solución se basó en la sustitución del amplificador de instrumentación por otro compatible eléctrica y mecánicamente y con mejores características de velocidad. El único inconveniente que tenía el componente sustituto era que no existía ningún modelo equivalente adecuado en librerías y repositorios de modelos de amplificadores de instrumentación. Por ello, se realizó un modelo comportamental del componente en LTSpice, a partir de la información que proporciona el fabricante en su hoja de características. Una vez realizado el modelo se simuló el circuito completo para verificar su correcto funcionamiento. El falso error desapareció con el amplificador de instrumentación seleccionado.

Como línea futura, se propone rediseñar los sistemas de enclavamientos rápidos y lentos con otro tipo de tecnología que permita una comunicación más fluida y una gobernanza más eficiente. En el RF test stand de ESS Bilbao los sensores y los sistemas de detección actualmente están basados en los circuitos de puertas lógicas rápidas. Estos son poco configurables. Podrían en un futuro estar comunicados entre sí, mediante tecnología embebida en FPGA's para las señales rápidas y mediante tecnología de PLC's en el caso de las señales lentas.

Respecto a las herramientas software, se desarrolló una interfaz gráfica de usuario (GUI) en MATLAB para automatización de medidas de potencia de RF de los klystron. Todos los valores obtenidos mediante dicha interfaz son muy significativos en la evaluación de las características electromagnéticas de los klystron. Con este programa, es posible postprocesar los datos y tener un registro de los diferentes amplificadores de alta potencia, en función de los números de serie. También se permite hacer un estudio de verificación y cumplimiento de los parámetros de diseño de los amplificadores klystron. Además, se puede configurar las condiciones de operación (anchos de pulso, frecuencias de repetición de los pulsos, tensión y corriente del modulador, potencia de entrada de la señal de RF). El otro tipo de medida que admite el software, a potencia de RF de entrada al klystron constante, variando la frecuencia de operación, es la respuesta frecuencial. Todas las gráficas y resultados se pueden almacenar por fecha y hora automáticamente facilitando la trazabilidad.

Como los resultados de las campañas de medidas de potencia se obtienen empleando distintas técnicas (detectores de potencia y calorimetría, además de métodos de analizador de espectros), es necesario un análisis que permita contrastar los datos y garantizar con certidumbre un rango acotado de valores de potencia, a pesar de la complejidad de la medida. Como línea futura de investigación, se propone en las medidas de alta potencia de RF del klystron, la aplicación de técnicas de *Machine Learning* para obtener una estimación del método de medición de potencia más verosímil, a partir de un modelo entrenado con múltiples datos tomados previamente. También que permita obtener el error cuadrático medio de cada conjunto de valores del estimador y que resuelva el problema de minimización del MSE (Mean Squared Error). Cuantos más datos tengamos para entrenar el modelo, más preciso será el modelo en la inferencia. Se puede implementar en MATLAB, R, Python o en cualquier otro lenguaje de programación. Otra posible mejora en la GUI de MATLAB desarrollada podría ser el

análisis y la caracterización de la forma del pulso de RF (duración del pulso, retraso, sobreimpulso, caída durante el pulso, rizado...).

Con respecto a la medición de alta potencia de RF mediante el método de calorimetría, se concluye que es apropiado cuando se dispone de un sistema de enfriamiento por agua u otros refrigerantes líquidos (siempre que se conozca su calor específico y su densidad). La medida de alta potencia de RF con el método de calorimetría permite obtener valores hasta megavatios de potencia de pico de RF de manera más exacta, en comparación con el método de detectores de potencia. Sobre la interfaz de usuario en LabView que se desarrolló para la medición, cabe destacar que es una herramienta que permite obtener el resultado de la medida de una manera muy visual y facilita el trabajo de reacondicionamiento de los klystron, sin necesidad de cálculos adicionales de potencia de RF.

Como futura tarea en la medida por calorimetría con la interfaz diseñada en LabView, se propone reducir las variaciones de las medidas de los sensores (caudalímetros y termopares) mediante el estudio de un prefiltrado de las muestras obtenidas con un filtro FIR diferente al que se implementó. Para ello, se podría resolver realizando diferentes promediados, la media móvil simple, ponderada o exponencial, e incorporar los valores de cada una de ellas, junto con su error cuadrático medio (MSE), en la interfaz de usuario.

## REFERENCIAS

- [1] <https://www.essbilbao.org/in-kind-contribution/rf-systems/>
- [2] R.W. Hamm, M. E. Hamm, “Industrial Accelerators and their Applications”, World Scientific Publishing, June 2012.
- [3] Ernest O. Lawrence and M. Stanley Livingston, “The Production of High Speed Light Ions Without the Use of High Voltages”, Physical Review 40, April 1932.
- [4] Kusch Jochen K, Amies C. Jude, “Linear Accelerator with X-ray Imaging Elements Mounting on Curved Support”, Siemens Medical Solutions USA, Inc., June 2004.
- [5] A. Caruso, F. Consoli, G. Gallo, D. Rifuggiato, E. Zappalà, A. Longhitano, M. Di Giacomo, “The LEBT Chopper for the Spiral2 Project”, Proceedings of IPAC2011, San Sebastián, Spain, September 2011.
- [6] J. W. Caldwell, Jr., “Fundamentals of Klystron Testing”, SLAC Technical Note TN 78-2, Stanford University, August 1978.
- [7] J.B. Lallement, V. Dimov, “LINACS”, Joint Universities Accelerator School JUAS-2016, Saint-Julien-en-Genevois, France, 2016.
- [8] T. Poggi, P. González, “A PCB Based Fast Interlock System for the Klystron”, ESS Bilbao Report, November 2015.
- [9] <https://www.minicircuits.com/pdfs/ZX47-40+.pdf> (Minicircuits Power Detector ZX47-40+ datasheet).
- [10] <https://www.analog.com/media/en/technical-documentation/datasheets/AD620.pdf> (Analog Devices Instrumentation amplifier AD620 datasheet).



- [11] <https://www.analog.com/media/en/technical-documentation/data-sheets/amp02.pdf> (Analog Devices High Accuracy Instrumentation Amplifier AMP02 datasheet).
- [12] T. C. Yu et al., “CPI 100KW Klystrons Operations Experiences in NSRRC”, Proceedings of IPAC2012, New Orleans, USA, 2012.
- [13] D. Christen, U. Badstuebner, J. Biela, and J.W. Kolar, “Calorimetric Power Loss Measurement for Highly Efficient Converters”, The 2010 International Power Electronics Conference - ECCE ASIA, Sapporo, Japan, 2010.
- [14] Erk Jensen, “Radiofrequency Cavities”, <https://cds.cern.ch/record/1997424?ln=es>, CERN, Geneva, Switzerland, 2012.
- [15] R. Miracoli, I. Arredondo, D. Belver, I. Bustinduy, J. Corres, P. Echevarria, M. Eguiraun, Z. Izaola, N. Garmendia, P. Gonzalez, L. Mugira, I. Rueda, “A Microwave Ion Source for Pulsed Proton Beam Production at ESS-Bilbao”, Proceedings of ECRIS-2014, Nizhny Novgorod, Russia, 2014.
- [16] S. H. Kim, “Protection of Accelerator Hardware: RF Systems”, Proceedings of the 2014 Joint International Accelerator School: Beam Loss and Accelerator Protection, Newport Beach, United States, November 2014.
- [17] Constantino Pérez Vega, “Amplificación de Alta Potencia de RF con Válvulas”, Dpto. de Ingeniería de Comunicaciones, Universidad de Cantabria, 2006.

## ANEXO I: Código fuente de la GUI para medidas del Klystron

A continuación, se presentarán las líneas del código de la GUI desarrollada en MATLAB descrito en el capítulo 4.

### Inicio del programa

```
function varargout = measurements(varargin)
% MEASUREMENTS M-file for measurements.fig
%     MEASUREMENTS, by itself, creates a new MEASUREMENTS or raises the existing
%     singleton*.
%
%     H = MEASUREMENTS returns the handle to a new MEASUREMENTS or the handle to
%     the existing singleton*.
%
%     MEASUREMENTS('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in MEASUREMENTS.M with the given input arguments.
%
%     MEASUREMENTS('Property','Value',...) creates a new MEASUREMENTS or raises
%     the existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before measurements_OpeningFunction gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to measurements_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help measurements
% Last Modified by GUIDE v2.5 15-Apr-2019 10:06:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @measurements_OpeningFcn, ...
                  'gui_OutputFcn',  @measurements_OutputFcn, ...
```

```
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);

if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before measurements is made visible.
function measurements_OpeningFcn(hObject, eventdata, handles, varargin)
% function list_box1_OpeningFcn(hObject, eventdata, handles, varargin)
%
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to measurements (see VARARGIN)

% Choose default command line output for measurements
handles.output = hObject;

guidata(hObject, handles);
% Update handles structure
%guidata(hObject, handles);

% UIWAIT makes measurements wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = measurements_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in connect.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function connect_Callback(hObject, eventdata, handles)%CONNECT
%Esta funcion es la de inicio de osciloscopio y configuracion

% hObject    handle to connect (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.i=1; %Inicializamos la variable asignada a 'Iteration'
set(handles.text_iteration,'String',num2str(handles.i)); %Presentamos dicha variable de Iteration en el GUI, este es el cuadro del iterador
%global oscilo;
% Cierra instrumentos (no se puede abrir un instrumento q no se ha cerrado previamente)
if ~(isempty(instrfind)) %Cerramos los instrumentos que haya abiertos
    fclose(instrfind);
end

tic

%handles.oscilo=oscilo;
handles.oscilo=visa('tek','GPIB8::1::INSTR'); %Inicializamos la variable 'handles.oscilo' para comunicarnos con el instrumento
handles.oscilo.Timeout=10.0;
handles.oscilo.inputBufferSize=42560;
handles.oscilo.EOIMode='on';
handles.oscilo.EOSMode='none';
fopen(handles.oscilo);
fprintf(handles.oscilo, 'ACquire:STATE RUN')
%COMPROBAMOS COMUNICACION
%Se realiza una verificacion de la correcta comunicacion
fprintf(handles.oscilo, '*IDN?');
idn = fscanf(handles.oscilo)
%fprintf(oscilo,'LOCK ALL') % Control modo remoto
%CLEAR Y RESET
fprintf(handles.oscilo,'*CLS')
fprintf(handles.oscilo,'*RST')
fprintf(handles.oscilo, 'ACquire:STATE RUN')
```

```
%MODULO DE ADQUISICION
fprintf(handles.oscilo,'ACQuire:MODE SAMple')    % Sin promediar
%CANALES ACTIVOS
fprintf(handles.oscilo,'SELECT:CH1 ON');
fprintf(handles.oscilo,'SELECT:CH2 ON');
fprintf(handles.oscilo,'SELECT:CH3 ON');
fprintf(handles.oscilo,'SELECT:CH4 ON');
% Realizamos un autoscale de cada canal x separado
fprintf(handles.oscilo,'AUTOSet EXECute')% Auto Scale para ajustar Time/Div y
Volt/Div
fprintf(handles.oscilo,['CH1:Offset 0'])
fprintf(handles.oscilo,['CH1:Probe 1'])%Sonda de medida 1x (atenuacion)
fprintf(handles.oscilo,['CH1:Position 4'])
fprintf(handles.oscilo,['CH2:Position -4'])
fprintf(handles.oscilo,['CH3:Position -4'])
fprintf(handles.oscilo,['CH4:Position -4'])
%TRIGGER
%Ajustes del Trigger, se fija la pendiente del trigger, el canal donde
%disparamos, y el nivel
fprintf(handles.oscilo, 'TRIGger:A:EDGE:SLOPe FALL');
fprintf(handles.oscilo, 'TRIGger:A:EDGE:SOUrce CH1');
fprintf(handles.oscilo, 'TRIGger:A:LEVel -1');
%FIJAMOS EL BW DE LOS CANALES
fprintf(handles.oscilo,'CH1:BANdwidth TWEnty');
fprintf(handles.oscilo,'CH2:BANdwidth TWEnty');
fprintf(handles.oscilo,'CH3:BANdwidth TWEnty');
fprintf(handles.oscilo,'CH4:BANdwidth TWEnty');
%ESCALAMOS LOS CANALES
fprintf(handles.oscilo, 'CH1:SCALE 1');
%fprintf(handles.oscilo, 'CH2:SCALE 1');
fprintf(handles.oscilo, 'CH2:SCALE 0.5');
fprintf(handles.oscilo, 'CH3:SCALE 0.5');
fprintf(handles.oscilo, 'CH4:SCALE 0.5');
%BASE DE TIEMPOS
fprintf(handles.oscilo, 'HORizontal:POSition 10');
fprintf(handles.oscilo, 'HORizontal:MODE:SCALE 1E-3');
fprintf(handles.oscilo, 'HORizontal:MODE:SCALE 0.5E-3');
%COMPROBADO HASTA AQUI %
fprintf(handles.oscilo,'ACQuire:MODE AVErAge')    % Promediado ON
fprintf(handles.oscilo,'ACQuire:STATE ON')    % Promediado ON
fprintf(handles.oscilo,'MEASURE:MEAS1:SOURCE CH1') %Seleccionamos la fuente de la
medida, en este caso la fuente de medida es el canal 1
fprintf(handles.oscilo,'MEASURE:MEAS1:TYPE AMP') %Aqui decimos que vamos a reali-
zar una medida de amplitud de la signal del canal 1, que es la que se activo
```

```
fprintf(handles.oscilo, 'MEASURE:MEAS1:STATE ON') %Iniciamos la medida
fprintf(handles.oscilo, 'MEASURE:MEAS2:SOURCE CH2') %Seleccionamos la fuente de medida en el canal 2
fprintf(handles.oscilo, 'MEASURE:MEAS2:TYPE LOW') %Le decimos que realice una medida de bajada
fprintf(handles.oscilo, 'MEASURE:MEAS2:STATE ON') %Activamos la medida
fprintf(handles.oscilo, 'MEASURE:MEAS3:SOURCE CH3') %Seleccionamos la fuente de medida en el canal 3
fprintf(handles.oscilo, 'MEASURE:MEAS3:TYPE LOW') %%Le decimos que realice una medida de bajada
fprintf(handles.oscilo, 'MEASURE:MEAS3:STATE ON') %Iniciamos la medida
fprintf(handles.oscilo, 'MEASURE:MEAS4:SOURCE CH4') %Seleccionamos la fuente de medida en el canal 4
fprintf(handles.oscilo, 'MEASURE:MEAS4:TYPE LOW') %Le decimos que realice una medida de bajada
fprintf(handles.oscilo, 'MEASURE:MEAS4:STATE ON') %Activamos la medida
fprintf(handles.oscilo, 'ACQUIRE:STATE ON') % Promediado ON
%fprintf(oscilo, '*OPC');
fprintf(handles.oscilo, '*WAI') %Le decimos que espere para tomar el valor, y este se estabilice
fprintf(handles.oscilo, 'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, 'MEASURE:MEAS1:mean?') %Pedimos el valor en la medida configurada como MEAS1, vista en la pantalla osciloscopio
handles.Vbeam_V=str2num(fscanf(handles.oscilo))%Variable de la medida del Vbeam
fprintf(handles.oscilo, 'MEASURE:MEAS2:mean?') %Pedimos el valor en la medida configurada como MEAS2, vista en la pantalla del osciloscopio
%%Codigo anadido el 03/Abril/2019
%Creamos las variables handles.Vklyrev, handles.Ibeam_V, Vdriverfwd_V, Vklyfwd_V
handles.Vklyrev_V=str2num(fscanf(handles.oscilo)) %Variable de la medida de V_klystron_reverse cuando la sonda V_klystron_reverse este conectada en el CH2, creamos la variable
fprintf(handles.oscilo, 'MEASURE:MEAS2:mean?') %Pedimos el valor en la medida configurada como MEAS2, vista en la pantalla del osciloscopio
handles.Ibeam_V=str2num(fscanf(handles.oscilo)) %Adquirimos el valor de la variable Ibeam cuando la sonda Ibeam este conectada en el CH2, creamos la variable
fprintf(handles.oscilo, 'MEASURE:MEAS3:mean?') %Pedimos el valor de la medida configurada como 'MEAS3' en la pantalla del osciloscopio
Vdriverfwd_V=str2num(fscanf(handles.oscilo))%Variable de la medida de V_driver_forward
fprintf(handles.oscilo, 'MEASURE:MEAS4:mean?') %Pedimos el valor de la medida configurada como 'MEAS4' en la pantalla del osciloscopio
Vklyfwd_V=str2num(fscanf(handles.oscilo))%Variable de la medida de V_klystron_forward
guidata(hObject, handles); %Guarda el estado de las variables 'handles'
whos
```

```
set(handles.text_fecha,'String',num2str(date)); %Mostramos la fecha en el GUIDE
guidata(hObject,handles);
%start(handles.tmr);

% Update handles structure
%guidata(hObject, handles);
guidata(hObject,handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\FIN FUNCION CONNECT\\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\\EMPIEZA FUNCION DE BOTON DISCONNECT\\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in disconnect.
function disconnect_Callback(hObject, eventdata, handles)%DISCONNECT

fclose(handles.oscilo)
close all
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\FIN FUNCION DISCONNECT\\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\\EMPIEZA FUNCION DE BOTON LOAD_CONF\\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in loadconf.
function loadconf_Callback(hObject, eventdata, handles)%LOAD_CONF
handles.i=1;%Inicializamos iteration
set(handles.text_iteration,'String',num2str(handles.i));%La mostramos en el GUI
guidata(hObject,handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Las siguientes 8 lineas de codigo son para borrar variables que hayan podido
quedar de los plot
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.z=plot(1,1);
clear handles.z
handles.Pdriver_W_meas = 0;
clear handles.Pdriver_W_meas;
handles.Pkly_KW_meas = 0;
clear handles.Pkly_KW_meas;
handles.Freq_MHz = 0;
clear handles.Freq_MHz;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%end
guidata(hObject,handles);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% A continuación, se carga el fichero del directorio
[archivo,ruta]=uigetfile('*.m','Abrir un archivo de datos');
if archivo == 0
    return;
else
    dat_archivo = strcat(ruta,archivo);
    selectedfile=fullfile(ruta,archivo);
    run(selectedfile);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Cargamos las variables del fichero
handles.Vbeam_KV_conf=Vbeam_KV;%Variable recogida vbeam del fichero de configura-
cion NOTA:Esta variable comienza por letra minuscuala 'vbeam'
handles.Ibeam_A=Ibeam_A;%Variable recogida ibeam del fichero de configuracion
handles.Freq_MHz=Freq_MHz; %Cargamos la variable de la frecuencia del fichero
handles.Repetition_frequency_Hz=Repetition_frequency_Hz; %Cargamos la frecuencia de
repeticion de pulso
handles.HV_pulse_duration_ms=HV_pulse_duration_ms; %Cargamos la duración de pulso
de HV
handles.RF_pulse_duration_ms=RF_pulse_duration_ms; %Cargamos la duración de pulso
de RF
handles.Pgen_dBm=Pgen_dBm; %Cargamos la frecuencia del generador de RF del fichero
handles.Pkly_KW_meas=zeros(1,length(handles.Freq_MHz)) %Inicializamos el vector de
la medida de potencia del Klystron
handles.measurement_type=measurement_type;
handles.C1_dB=C1_dB; %variable de acoplamiento 1
handles.Slope1_V_dB=Slope1_V_dB; %variable de la pendiente del detector 1
handles.Vo1_V=Vo1_V; %variable de tensión relacionada con el detector
handles.Lcable_dB=Lcable_dB; %perdidas del cable
handles.C2_dB=C2_dB; %variable de acoplamiento 2
handles.Slope2_V_dB=Slope2_V_dB;%variable de la pendiente del detector 2
handles.Vo2_V=Vo2_V;%variable de tensión relacionada con el detector 2
```



```
handles.klystron_model=Klystron_model;
handles.klystron_serial_number=Klystron_serial_number;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(handles.Ibeam_button,'value',1); %Fijamos el tick de Ibeam por defecto a 1
handles.channel2_signal = 'Ibeam'
str2= 'Vbeam(V)      Ibeam(V)      VdetPdriver(V)  VdetPklyfwd(V)
Pgen(dBm)      Pdriver(W)      Pkly(KW)      Vbeam(KV)      Ibeam(V)
Ibeam(A)';
set(handles.Tabla_valores_capturados,'String',str2)
set(handles.Pklyrv_button,'value',0); %Fijamos el tick de Pklyrv_button por defecto
en el frontpanel a 0
if handles.measurement_type == 'PinVsPout' %Si la medida es de tipo PinVsPout
    set(handles.text_titulo_grafica,'String','Pin Vs. Pout') %Representamos los va-
lores en los textboxes
    set(handles.text_titulo_Pgen_o_freq_inf,'String','Pgen(dBm)');
    set(handles.text_pgen_o_freq_tab_sup,'String','Frequency(MHz)');
    set(handles.text_Freq_tabla_sup,'String',num2str(handles.Freq_MHz));%Muestro la
frecuencia en el guide
    set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm(handles.i)));
    set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(han-
dles.i)));%Muestro en el guide la potencia de RF
    handles.N=length(Pgen_dBm)%Variable que indica la longitud del vector de poten-
cia de generador de RF
else
    handles.N=length(Freq_MHz)
    handles.i=1;
    set(handles.text_titulo_grafica,'String','FREQUENCY MEASUREMENT')
    set(handles.text_Freq_tabla_sup,'String',num2str(handles.Pgen_dBm));%Muestro la
potencia del generador en el guide
    set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm));
    set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Freq_MHz(hand-
les.i)));%Muestro en el guide la potencia de RF
    set(handles.text_pgen_o_freq_tab_sup,'String','Pgen(dBm)');
    set(handles.text_titulo_Pgen_o_freq_inf,'String','Frequency(MHz)');

end
set(handles.text_modelo_klystron,'String',handles.klystron_model);
set(handles.text_klystron_serial_number,'String',handles.klystron_serial_number);
set(handles.text_Vbeam_tab_sup,'String',num2str(handles.Vbeam_KV_conf));%Muestro en
el guide la variable Vbeam
set(handles.text_Ibeam_tab_sup,'String',num2str(handles.Ibeam_A));
set(handles.text_PRF,'String',num2str(handles.Repetition_frequency_Hz));%Muestro la
frecuencia de repeticion de pulso en el guide
```

```

set(handles.text_HV_pulse,'String',num2str(handles.HV_pulse_duration_ms));%Muestro
la duracion del pulso de HV en el guide
set(handles.text_RF_pulse,'String',num2str(handles.RF_pulse_duration_ms));%Muestro
la frecuencia en el guide

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIMER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.tmr = timer(...
    'ExecutionMode', 'fixedRate', ...
    'Period', 0.3, ...
    'Busymode', 'queue', ...
    'TimerFcn', {@timerCallback,handles});
guidata(hObject,handles);
start(handles.tmr); %Iniciamos el timer para representar los 4 canales dinámica-
mente
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN TIMER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Inicializamos las variables:
% handles.Vklyrev=zeros(1,handles.N);
% handles.Ibeam_V=zeros(1,handles.N);
% handles.Vbeam_V=zeros(1,handles.N);
% handles.Vbeam_KV_meas=zeros(1,handles.N);
% handles.A=zeros(handles.N,8);%Inicializamos matriz de salida de datos para el ar-
chivo que exportaremos al EXCEL
% handles.Vklyfwd_V=zeros(1,handles.N);
% handles.result=zeros(handles.N,9);
% handles.result2=zeros(handles.N,9);
% handles.result3=zeros(handles.N,10);
% handles.Vdriverfwd_V=zeros(1,handles.N);
% %handles.Vbeam_KV_meas=zeros(1,handles.N);
% %handles.Pdriver_dBm_meas=zeros(1,handles.N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\\FIN FUNCION LOAD_CONF\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FUNCION DE INTERRUPCION POR TIMER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function timerCallback(hObject, eventdata, handles)
% function timerCallback(hObject, eventdata, handles)
% %handles = guidata(hObject);
% % do some processing here
% %guidata(hObject,handles);

```

## ANEXO I: Código fuente de la GUI para medidas del Klystron

```
#####En los textboxes mostramos los valores de las medidas de los 4 canales dinámi-
camente
fprintf(handles.oscilo,'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, '*WAI')
fprintf(handles.oscilo,'MEASURE:MEAS1:mean?')
ch1=str2num(fscanf(handles.oscilo));
set(handles.text_Vbeam_tabla,'String',num2str(ch1))
fprintf(handles.oscilo,'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, '*WAI')
fprintf(handles.oscilo,'MEASURE:MEAS2:mean?')
ch2=str2num(fscanf(handles.oscilo));
set(handles.text_Ibeam_Pklyr,'String',num2str(ch2))
fprintf(handles.oscilo,'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, '*WAI')
fprintf(handles.oscilo,'MEASURE:MEAS3:mean?')
ch3=str2num(fscanf(handles.oscilo));
set(handles.text_VdetPdriver,'String',num2str(ch3))
fprintf(handles.oscilo,'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, '*WAI')
fprintf(handles.oscilo,'MEASURE:MEAS4:mean?')
ch4=str2num(fscanf(handles.oscilo));
set(handles.text_Pklyforward,'String',num2str(ch4))

end

#####FIN FUNCION DE INTERRUPCION POR TIMER#####

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\\EMPIEZA FUNCION DE BOTON CAPTURE\\\\\\\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Cada vez que ejecutamos Capture adquiere nuevos valores de las variables
% --- Executes on button press in capture.
function capture_Callback(hObject, eventdata, handles)%CAPTURE
Navg = 16
whos
%23 de Abril%
stop(handles.tmr)
fprintf(handles.oscilo,['ACQuire:NUMAVg ',int2str(Navg)]) % Promediado (aqui no le
activamos)
fprintf(handles.oscilo,'ACQuire:MODE AVErage') % Promediado ON
fprintf(handles.oscilo,'ACQuire:STATE ON') % Promediado ON
%fprintf(handles.oscilo,'*OPC');
```

```
fprintf(handles.oscilo, '*WAI')
    fprintf(handles.oscilo, 'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, 'MEASURE:MEAS1:mean?')
handles.Vbeam_V(1,handles.i)=str2num(fscanf(handles.oscilo))
disp('Mostrando la variable handles.Vbeam')
handles.Vbeam_V
%%Codigo anadido el 03/Abril/2019
fprintf(handles.oscilo, 'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, 'MEASURE:MEAS2:mean?')
if handles.channel2_signal == 'Pklyr' %Variable que indica si esta pulsado el Pkly-
rev (1) o Ibeam (0)
    handles.Vklyrev_V(1,handles.i)=str2num(fscanf(handles.oscilo))%Variable de la
medida de V_klystron_reverse
    set(handles.text_Ibeam_Pklyr, 'String', num2str(handles.Vklyrev_V(1,handles.i)));
else
    handles.Ibeam_V(1,handles.i)=str2num(fscanf(handles.oscilo))
    set(handles.text_Ibeam_Pklyr, 'String', num2str(handles.Ibeam_V(1,handles.i)));
    set(handles.text_Ibeam_tabla, 'String', num2str(handles.Ibeam_V(1,handles.i)));
    disp('Mostrando las variables handles.Ibeam_V')
    handles.Ibeam_V(1,handles.i)
end
guidata(hObject,handles);
disp('Vklyrev_V y Ibeam_V')
handles.Vklyrev_V
handles.Ibeam_V
    fprintf(handles.oscilo, 'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, 'MEASURE:MEAS3:mean?')
handles.Vdriverfwd_V(1,handles.i)=str2num(fscanf(handles.oscilo))
    fprintf(handles.oscilo, 'MEASUREMENT:STATISTICS:COUNT RESET');
fprintf(handles.oscilo, 'MEASURE:MEAS4:mean?')
handles.Vklyfwd_V(1,handles.i)=str2num(fscanf(handles.oscilo))

%set(handles.text_Vbeam_tabla, 'String', num2str(handles.Vbeam_V(1,handles.i)));
%set(handles.text_Pklyforward, 'String', num2str(handles.Vklyfwd_V(1,handles.i)));
set(handles.text_VdetPdriver, 'String', num2str(handles.Vdriverfwd_V(1,handles.i)));
handles.Vbeam_KV_meas(1,handles.i)=handles.Vbeam_V(1,handles.i)*15; % Calculamos la
tensión del modulador
set(handles.text_Vbeam_tabla_medidas, 'String', num2str(handles.Vbeam_KV_meas(1,han-
dles.i)))
set(handles.text_Vbeam_tabla_medidas, 'String', num2str(handles.Vbeam_KV_meas(1,han-
dles.i)))
disp('Mostrando las variables handles.Vdriverfwd_V')
handles.Vdriverfwd_V(1,handles.i)
```

```
handles.Pdriver_dBm_meas(1,handles.i)=(handles.Vdriverfwd_V(1,handles.i)-han-
dles.Vo1_V)/(handles.Slope1_V_dB)%variable vector fila que representa la potencia
del driver en dBm
handles.Pdriver_W_meas(1,handles.i)=10.^((handles.Pdriver_dBm_meas(1,handles.i)-
30+handles.C1_dB)./10); %variable que representa la potencia del driver en Watts
handles.Pdriver_dBm_meas
handles.Pdriver_W_meas
disp('Mostrando la variable handles.Pdriver_dBm_meas ')
handles.Pdriver_dBm_meas
handles.Pkly_dBm_meas(1,handles.i)=(handles.Vklyfwd_V(1,handles.i)-han-
dles.Vo2_V)/(handles.Slope2_V_dB)%variable que representa la potencia del driver en
dBm
handles.Pkly_KW_meas(1,handles.i)=10.^((handles.Pkly_dBm_meas(1,handles.i)-60+han-
dles.C2_dB)./10) %variable que representa la potencia del klystron en KiloWatts
handles.Pkly_dBm_meas
handles.Pkly_KW_meas
disp('Mostrando las variables handles.Pkly_dBm_meas(1,handles.i)...')
handles.Pkly_dBm_meas(1,handles.i)
handles.Pkly_KW_meas(1,handles.i)
handles.Vklyfwd_V(1,handles.i)
guidata(hObject,handles);
%Matriz que se exporta los datos al fichero de EXCEL, el nombre de dicha
%matriz es 'handles.A'

if handles.measurement_type == 'PinVsPout' %si estamos en la medida Pin vs Pout
    if handles.channel2_signal == 'Pklyr' % CH2 Pklyrev
        handles.A(handles.i,:)= [handles.Vbeam_V(1,handles.i) han-
dles.Vklyrev_V(1,handles.i) handles.Vdriverfwd_V(1,handles.i)...
handles.Vklyfwd_V(1,handles.i) handles.Pgen_dBm(1,handles.i) han-
dles.Pdriver_W_meas(1,handles.i) ...
handles.Pkly_KW_meas(1,handles.i) handles.Vbeam_KV_meas(1,handles.i)
]
        %set(handles.text_Ibeam_Pklyr,'String', num2str(handles.Vklyrev_V(1,han-
dles.i)));
        set(handles.text_Ibeam_tabla,'String','---');
        set(handles.text_Ibeam_Amperios,'String','---');
    else
        handles.result(handles.i,:)= [handles.Vbeam_V(1,handles.i) han-
dles.Ibeam_V(1,handles.i) handles.Vdriverfwd_V(1,handles.i)...
handles.Vklyfwd_V(1,handles.i) handles.Pgen_dBm(1,handles.i) han-
dles.Pdriver_W_meas(1,handles.i) ...
handles.Pkly_KW_meas(1,handles.i) handles.Vbeam_KV_meas(1,handles.i)
10*handles.Ibeam_V(1,handles.i)]
        %set(handles.text_Ibeam_Pklyr,'String', num2str(handles.Ibeam_V(1,han-
dles.i)));
```

```

        set(handles.text_Ibeam_Amperios,'String', num2str(10*handles.Ibeam_V(1,handles.i)));
        handles.result
    end
else %%Si estamos en la medida frecuencial
    if handles.channel2_signal == 'Pklyr'
        handles.result2(handles.i,:)=[handles.Freq_MHz(1,handles.i) handles.Vbeam_V(1,handles.i) handles.Vklyrev_V(1,handles.i)...
            handles.Vdriverfwd_V(1,handles.i) handles.Vklyfwd_V(1,handles.i) handles.Pgen_dBm handles.Pdriver_W_meas(1,handles.i)...
            handles.Pkly_KW_meas(1,handles.i) handles.Vbeam_KV_meas(1,handles.i)
        ]
        %set(handles.text_Ibeam_Pklyr,'String', num2str(handles.Vklyrev_V(1,handles.i)));
        set(handles.text_Ibeam_tabla,'String','---');
        set(handles.text_Ibeam_Amperios,'String','---');
    else
        handles.result3(handles.i,:)=[handles.Freq_MHz(1,handles.i) handles.Vbeam_V(1,handles.i) handles.Ibeam_V(1,handles.i) ...
            handles.Vdriverfwd_V(1,handles.i) handles.Vklyfwd_V(1,handles.i) handles.Pgen_dBm handles.Pdriver_W_meas(1,handles.i)...
            handles.Pkly_KW_meas(1,handles.i) handles.Vbeam_KV_meas(1,handles.i) 10*handles.Ibeam_V(1,handles.i) ]
        %set(handles.text_Ibeam_Pklyr,'String', num2str(handles.Ibeam_V(1,handles.i)));
        set(handles.text_Ibeam_Amperios,'String', num2str(10*handles.Ibeam_V(1,handles.i)));
    end
end
set(handles.text_Pdriver,'String',num2str(handles.Pdriver_W_meas(1,handles.i)));
set(handles.text_Pklystron,'String',num2str(handles.Pkly_KW_meas(1,handles.i)));
size(handles.Pkly_KW_meas)
size(handles.Freq_MHz)
guidata(hObject,handles);
% PLOT
if handles.measurement_type == 'PinVsPout' %Indica que la medida es de tipo Pin Vs. Pout
    handles.Pdriver_W_meas
    handles.Pkly_KW_meas
    handles.z=plot(handles.Pdriver_W_meas,handles.Pkly_KW_meas,'r-o')
    set(handles.text_titulo_grafica,'String','Pin Vs. Pout')
    xlabel('Pdriver(W)')
    ylabel('Pklystron(KW)')
    grid on; zoom on;
end

```

```

guidata(hObject,handles);
set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(1,handles.i)));
set(handles.text_Freq_tabla_sup,'String',num2str(handles.Freq_MHz));
else % Aqui seria medida de tipo frecuencial.
handles.z=plot(handles.Freq_MHz,handles.Pkly_KW_meas,'b-o')
set(handles.text_titulo_grafica,'String','FREQUENCY MEASUREMENT')
xlabel('Frequency(MHz)')
ylabel('Pklystron(KW)')
grid on; zoom on;
guidata(hObject,handles);
set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Freq_MHz(1,handles.i)));
set(handles.text_Freq_tabla_sup,'String',num2str(handles.Pgen_dBm));
guidata(hObject,handles);
end
start(handles.tmr)
guidata(hObject,handles);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in complete.
function complete_Callback(hObject, eventdata, handles)%%COMPLETE
whos %Muestra los valores de todas las variables
%%Con 'getframe' Capturamos la pantalla del GUI
stop(handles.tmr)
f=getframe(gcf);
[X,Map]=frame2im(f);
h=clock;
h=fix(h(4:6));
texto=['_',num2str(date),'_',num2str(h(1)),'-',num2str(h(2)),'-',num2str(h(3))];
imwrite(X,['Klystron_Measurements_',texto,'.jpg'])
%%Y la guardamos como .jpg guardamos
%Exportamos el fichero en la ruta especificada, en este caso en el mismo directorio
%donde se almacena el GUI
curr_dir=cd

```

```
[fichero,directorio]=uiputfile('*.csv','Salvando Resultados');
cd(directorio);

filenamelength=length(fichero);
if (filenamelength<=4) | (fichero(filenamelength-3:filenamelength)~='.csv')
    disp('No se llama .csv')
    fichero=[fichero, '.csv'];
end

notas='No';
h=measurements;
% ESCRITURA DE UN FICHERO CSV CON CABECERA DE CONDICIONES DE MEDIDA
if handles.measurement_type == 'PinVsPout'
    if handles.channel2_signal == 'Ibeam'
        fid=fopen(fichero,'wt'); % apertura del fichero (escritura-texto)
        fprintf(fid,'Authors, Alvaro \n');
        fprintf(fid,'Date, %s\n', date);
        hora=clock;
        fprintf(fid,'Time, %d h %d min %d sec\n', fix(hora(4:6)));
        fprintf(fid,'Comments, %s\n', notas);
        fprintf(fid,'Klystron Model, %s\n',handles.klystron_model);
        fprintf(fid,'Klystron S/N, %s\n',handles.klystron_serial_number);
        fprintf(fid,'Vbeam(KV),%4.1f\n',handles.Vbeam_KV_conf);
        fprintf(fid,'Ibeam(A),%4.1f\n',handles.Ibeam_A);
        fprintf(fid,'freq(MHz),%4.3f\n',handles.Freq_MHz);
        fprintf(fid,'HV_pulse_duration(ms),%4.2f\n',handles.HV_pulse_duration_ms);
        fprintf(fid,'RF_pulse_duration(ms),%4.2f\n',handles.RF_pulse_duration_ms);
        fprintf(fid,'Repetition_frequency(Hz),%4.1f\n',handles.Repetition_fre-
quency_Hz);
        fprintf(fid,'C1(dB),%4.2f\n',handles.C1_dB);
        fprintf(fid,'Slope1(V/dB),%4.4f\n',handles.Slope1_V_dB);
        fprintf(fid,'Vol(V),%4.4f\n',handles.Vol_V);
        fprintf(fid,'Lcable(dB),%4.2f\n',handles.Lcable_dB);
        fprintf(fid,'C2,%4.3f\n',handles.C2_dB);
        fprintf(fid,'Slope2(V/dB),%4.4f\n',handles.Slope2_V_dB);
        fprintf(fid,'Vo2(V),%4.4f\n',handles.Vo2_V);
        fprintf(fid,'Vbeam(V), Ibeam(V), Vdriver(V), Vkly(V), Pgen(dBm),
Pdriver(W), Pkly(KW), Vbeam(KV)_meas, Ibeam(A)\n')
        D=handles.result'; %%Esta es la matriz D de resultados para escribir en la
tabla
        %%por columnas
        fprintf(fid,'%4.2f, %4.3f, %4.4f, %4.3f, %4.1f, %4.2f, %4.3f, %4.3f,
%4.3f\n',D);
    else
```



```

    fid=fopen(fichero,'wt'); % apertura del fichero (escritura-texto)
    fprintf(fid,'Authors, Alvaro \n');
    fprintf(fid,'Date, %s\n', date);
    hora=clock;
    fprintf(fid,'Time, %d h %d min %d sec\n', fix(hora(4:6)));
    fprintf(fid,'Comments, %s\n', notas);
    fprintf(fid,'Klystron Model, %s\n',handles.klystron_model);
    fprintf(fid,'Klystron S/N, %s\n',handles.klystron_serial_number);
    fprintf(fid,'Vbeam(KV),%4.1f\n',handles.Vbeam_KV_conf);
    fprintf(fid,'Ibeam(A),%4.1f\n',handles.Ibeam_A);
    fprintf(fid,'freq(MHz),%4.3f\n',handles.Freq_MHz);
    fprintf(fid,'HV_pulse_duration(ms),%4.2f\n',handles.HV_pulse_duration_ms);
    fprintf(fid,'RF_pulse_duration(ms),%4.2f\n',handles.RF_pulse_duration_ms);
    fprintf(fid,'Repetition_frequency(Hz),%4.1f\n',handles.Repetition_fre-
quency_Hz);

    fprintf(fid,'Cl(dB),%4.2f\n',handles.Cl_dB);
    fprintf(fid,'Slope1(V/dB),%4.4f\n',handles.Slope1_V_dB);
    fprintf(fid,'Vol(V),%4.4f\n',handles.Vol_V);
    fprintf(fid,'Lcable(dB),%4.2f\n',handles.Lcable_dB);
    fprintf(fid,'C2,%4.3f\n',handles.C2_dB);
    fprintf(fid,'Slope2(V/dB),%4.4f\n',handles.Slope2_V_dB);
    fprintf(fid,'Vo2(V),%4.4f\n',handles.Vo2_V);
    fprintf(fid,'Vbeam(V), Pdetklyrev(V), Vdriver(V), Vkly(V), Pgen(dBm),
Pdriver(W), Pkly(KW), Vbeam(KV)_meas\n')

    D=handles.A';%%Esta es la matriz D de resultados para escribir en la tabla
    %%por columnas
    fprintf(fid,'%4.2f, %4.3f, %4.4f, %4.3f, %4.1f, %4.2f, %4.3f, %4.3f\n',D);

end
else
    if handles.channel2_signal == 'Pklyr'
        fid=fopen(fichero,'wt'); % apertura del fichero (escritura-texto)
        fprintf(fid,'Authors, Alvaro \n');
        fprintf(fid,'Date, %s\n', date);
        hora=clock;
        fprintf(fid,'Time, %d h %d min %d sec\n', fix(hora(4:6)));
        fprintf(fid,'Comments, %s\n', notas);
        fprintf(fid,'Klystron Model, %s\n',handles.klystron_model);
        fprintf(fid,'Klystron S/N, %s\n',handles.klystron_serial_number);
        fprintf(fid,'Vbeam(KV),%4.1f\n',handles.Vbeam_KV_conf);
        fprintf(fid,'Ibeam(A),%4.1f\n',handles.Ibeam_A);
        fprintf(fid,'HV_pulse_duration(ms),%4.2f\n',handles.HV_pulse_duration_ms);
        fprintf(fid,'RF_pulse_duration(ms),%4.2f\n',handles.RF_pulse_duration_ms);

```

```

    fprintf(fid, 'Repetition_frequency(Hz), %4.1f\n', handles.Repetition_fre-
quency_Hz);
    fprintf(fid, 'C1(dB), %4.2f\n', handles.C1_dB);
    fprintf(fid, 'Slope1(V/dB), %4.4f\n', handles.Slope1_V_dB);
    fprintf(fid, 'Vol(V), %4.4f\n', handles.Vol_V);
    fprintf(fid, 'Lcable(dB), %4.2f\n', handles.Lcable_dB);
    fprintf(fid, 'C2, %4.3f\n', handles.C2_dB);
    fprintf(fid, 'Slope2(V/dB), %4.4f\n', handles.Slope2_V_dB);
    fprintf(fid, 'Vo2(V), %4.4f\n', handles.Vo2_V);
    fprintf(fid, 'Freq_MHz, Vbeam(V), PdetKlyrev(V), Vdriver(V), Vkly(V),
Pgen(dBm), Pdriver(W), Pkly(KW), Vbeam(KV)_meas\n')
    D=handles.result2'; %%Esta es la matriz D de resultados para escribir en la
tabla
    %%por columnas
    fprintf(fid, '%4.2f, %4.3f, %4.4f, %4.3f, %4.1f, %4.2f, %4.3f, %4.3f,
%4.3f\n', D);
else
    fid=fopen(fichero, 'wt'); % apertura del fichero (escritura-texto)
    fprintf(fid, 'Authors, Alvaro \n');
    fprintf(fid, 'Date, %s\n', date);
    hora=clock;
    fprintf(fid, 'Time, %d h %d min %d sec\n', fix(hora(4:6)));
    fprintf(fid, 'Comments, %s\n', notas);
    fprintf(fid, 'Vbeam(KV), %4.1f\n', handles.Vbeam_KV_conf);
    fprintf(fid, 'Ibeam(A), %4.1f\n', handles.Ibeam_A);
    fprintf(fid, 'HV_pulse_duration(ms), %4.2f\n', handles.HV_pulse_duration_ms);
    fprintf(fid, 'RF_pulse_duration(ms), %4.2f\n', handles.RF_pulse_duration_ms);
    fprintf(fid, 'Repetition_frequency(Hz), %4.1f\n', handles.Repetition_fre-
quency_Hz);
    fprintf(fid, 'C1(dB), %4.2f\n', handles.C1_dB);
    fprintf(fid, 'Slope1(V/dB), %4.4f\n', handles.Slope1_V_dB);
    fprintf(fid, 'Vol(V), %4.4f\n', handles.Vol_V);
    fprintf(fid, 'Lcable(dB), %4.2f\n', handles.Lcable_dB);
    fprintf(fid, 'C2, %4.3f\n', handles.C2_dB);
    fprintf(fid, 'Slope2(V/dB), %4.4f\n', handles.Slope2_V_dB);
    fprintf(fid, 'Vo2(V), %4.4f\n', handles.Vo2_V);
    fprintf(fid, 'Freq_MHz, Vbeam(V), Ibeam(V), Vdriver(V), Vkly(V), Pgen(dBm),
Pdriver(W), Pkly(KW), Vbeam(KV)_meas, Ibeam(A)\n')
    D=handles.result3'; %%Esta es la matriz D de resultados para escribir en la
tabla
    %%por columnas
    fprintf(fid, '%4.2f, %4.3f, %4.4f, %4.3f, %4.3f, %4.1f, %4.2f, %4.3f, %4.3f,
%4.4f\n', D);
end

```

```

end
fclose(fid); % cierre del fichero
cd(curr_dir);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in previous.
function previous_Callback(hObject, eventdata, handles)%%PREVIOUS
% hObject    handle to previous (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if handles.measurement_type == 'PinVsPout'
    set(handles.text_iteration,'String',num2str(handles.i));
    set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(handles.i)));
    set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm(handles.i)));
    handles.i
    if handles.i >= 2
        handles.i=handles.i-1;
        set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(handles.i)));
        set(handles.text_iteration,'String',num2str(handles.i));
        set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm(handles.i)));
        handles.i
    end
end
if handles.measurement_type == 'Frequency'
    handles.measurement_type
    if handles.i >= 2
        handles.i=handles.i-1;
        set(handles.text_iteration,'String',num2str(handles.i));
        set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Freq_MHz(1,handles.i)));
        handles.i
        %handles.measurement_type
    end
end

```

```

end
end
guidata(hObject,handles);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\\FIN FUNCION PREVIOUS\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\\EMPIEZA LA FUNCION DEL BOTON NEXT\\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in next.
function next_Callback(hObject, eventdata, handles)%%NEXT

if handles.measurement_type == 'PinVsPout'
    set(handles.text_iteration,'String',num2str(handles.i));
    set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(handles.i)));
    set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm(handles.i)));
    handles.i
    if handles.i < handles.N
        handles.i=handles.i+1;
        set(handles.text_iteration,'String',num2str(handles.i));
        set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Pgen_dBm(handles.i)));
        set(handles.text_Pgen_tabla,'String',num2str(handles.Pgen_dBm(handles.i)));
        handles.i
    end
else
    if handles.i < handles.N
        handles.i=handles.i+1;
        set(handles.text_iteration,'String',num2str(handles.i));
        set(handles.text_Pgen_drcha_iteration,'String',num2str(handles.Freq_MHz(1,handles.i)));
        handles.i
    end
end
guidata(hObject,handles);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\FIN FUNCION NEXT\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in Ibeam_button.
function Ibeam_button_Callback(hObject, eventdata, handles)

handles.a=get(handles.Ibeam_button,'value')
if handles.a == 1
    handles.channel2_signal = 'Ibeam' % La variable 'handles.channel2_signal' indica 0 o 1 dependiendo de si queremos medir Ibeam o Pklyrev
    set(handles.Pklyrv_button,'value',0)
    str2= 'Vbeam(V)      Ibeam(V)              VdetPdriver(V)  VdetPklyfwd(V)
Pgen(dBm)      Pdriver(W)          Pkly(KW)          Vbeam(KV)      Ibeam(V)
Ibeam(A)';
    set(handles.Tabla_valores_capturados,'String',str2)
else
    handles.channel2_signal = 'Pklyr'
end
guidata(hObject,handles);

% --- Executes on button press in Pklyrv_button.
function Pklyrv_button_Callback(hObject, eventdata, handles)

handles.a=get(handles.Pklyrv_button,'value')
if handles.a == 1
    handles.channel2_signal = 'Pklyr'
    str2= 'Vbeam(V)      VdetPklyrev(V)      VdetPdriver(V)  VdetPklyfwd(V)
Pgen(dBm)      Pdriver(W)          Pkly(KW)          Vbeam(KV)      Ibeam(V)
Ibeam(A)';
    set(handles.Tabla_valores_capturados,'String',str2)
    set(handles.Ibeam_button,'value',0)
else
    handles.channel2_signal = 'Ibeam'
end

handles.a=get(handles.Pklyrv_button,'value')
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```
% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

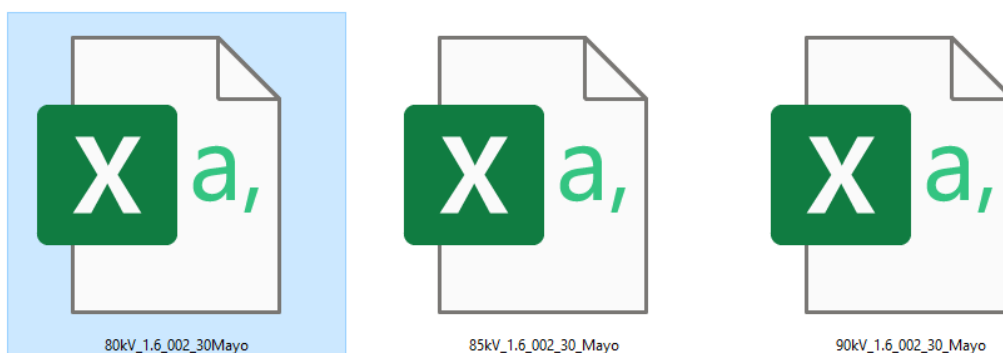
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

## ANEXO II: Procesado del fichero de resultados de la GUI de MATLAB

A continuación, se presenta un ejemplo del fichero de resultados generado por el botón COMPLETE del GUI desarrollado y descrito en el capítulo 4, que podrá ser interpretado posteriormente por la hoja de cálculo Microsoft EXCEL, o cualquier otro tipo de software compatible con la extensión “.csv”.

Para su representación en Microsoft EXCEL, se explicarán una serie de pasos previos necesarios para una visualización más amigable y práctica de los resultados.

- En primer lugar seleccionamos el documento .csv que queramos visualizar.



## ANEXO II: Procesado del fichero de resultados de la GUI de MATLAB

- Lo abrimos sobre el software de hoja de cálculo.

	A	B	C	D	E	F	G	H
1	Authors, Alvaro							
2	Date, 30-May-2019							
3	Time, 12 h 22 min 44 sec							
4	Comments, No							
5	Klystron Model, VKP-8352A							
6	Klystron S/N, 002							
7	Vbeam(KV), 80.0							
8	Ibeam(A), 41.0							
9	freq(MHz), 352.210							
10	HV_pulse_duration(ms), 1.60							
11	RF_pulse_duration(ms), 1.40							
12	Repetition_frequency(Hz), 14.0							
13	C1(dB), 60.00							
14	Slope1(V/dB), -0.0250							
15	Vo1(V), 1.0792							
16	Lcable(dB), 0.70							
17	C2, 83.700							
18	Slope2(V/dB), -0.0244							
19	Vo2(V), 1.0844							
20	Vbeam(V), Ibeam(V), Vdriver(V), Vkly(V), Pgen(dBm), Pdriver(W), Pkly(KW), Vbeam(KV)_meas, Ibeam(A)							
21	-5.6450, -2.9358, 1.6936, 1.2754, -10.00, 3.5, 38.6, -84.7, -29.4							
22	-5.6450, -2.9456, 1.6494, 1.2308, -9.00, 5.2, 58.9, -84.7, -29.5							
23	-5.6425, -2.9638, 1.6098, 1.1931, -8.00, 7.5, 84.0, -84.6, -29.6							
24	-5.6383, -2.9167, 1.5640, 1.1553, -7.00, 11.5, 120.1, -84.6, -29.2							
25	-5.6457, -2.9478, 1.5278, 1.1214, -6.00, 16.1, 165.4, -84.7, -29.5							
26	-5.6417, -2.9258, 1.4963, 1.1048, -5.00, 21.5, 193.4, -84.6, -29.3							
27	-5.6363, -2.9469, 1.4359, 1.0484, -4.00, 37.4, 329.1, -84.5, -29.5							
28	-5.6463, -2.9500, 1.3997, 1.0194, -3.00, 52.2, 433.0, -84.7, -29.5							
29	-5.6367, -2.9467, 1.3655, 0.9931, -2.00, 71.6, 554.7, -84.6, -29.5							
30	-5.6375, -2.9350, 1.3506, 0.9819, -1.50, 82.1, 616.9, -84.6, -29.4							
31	-5.6417, -2.9558, 1.3342, 0.9692, -1.00, 95.5, 695.5, -84.6, -29.6							
32	-5.6425, -2.9581, 1.3175, 0.9537, -0.50, 111.4, 804.4, -84.6, -29.6							
33	-5.6383, -2.9083, 1.3010, 0.9387, 0.00, 129.6, 926.7, -84.6, -29.1							
34	-5.6367, -2.9175, 1.2829, 0.9241, 0.50, 153.2, 1064.4, -84.6, -29.2							
35	-5.6383, -2.9467, 1.2696, 0.9171, 1.00, 173.2, 1136.9, -84.6, -29.5							
36	-5.6400, -2.9275, 1.2546, 0.9087, 1.50, 198.8, 1229.9, -84.6, -29.3							
37	-5.6300, -2.9317, 1.2409, 0.9043, 2.00, 225.6, 1283.3, -84.5, -29.3							
38	-5.6400, -2.9500, 1.2305, 0.8975, 2.50, 248.3, 1367.7, -84.6, -29.5							
39	-5.6383, -2.9433, 1.2185, 0.8962, 3.00, 277.1, 1383.9, -84.6, -29.4							

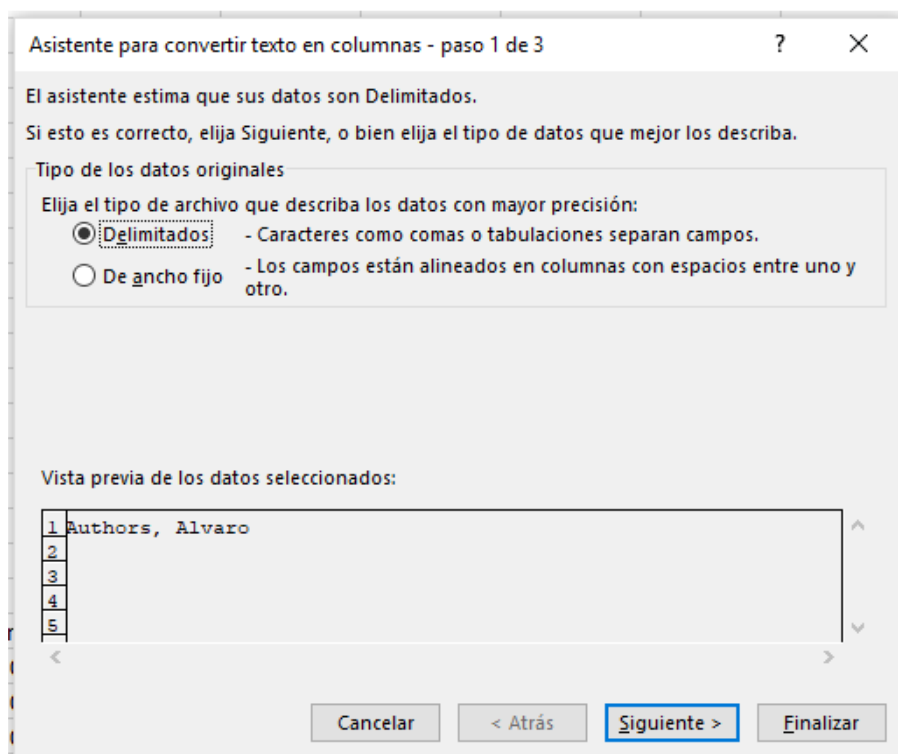
- Observamos que los datos se encuentran en una misma celda, lo que interesa para procesarlos, es separarlos por celdas, y así manejarlos más prácticamente. Para ello, seleccionamos sólo la primera columna, desde el primer dato hasta el último que tenga, posteriormente en la parte superior, en la barra de herramientas, seleccionamos la pestaña ‘datos’, y dentro de ella seleccionamos ‘texto en columnas’:



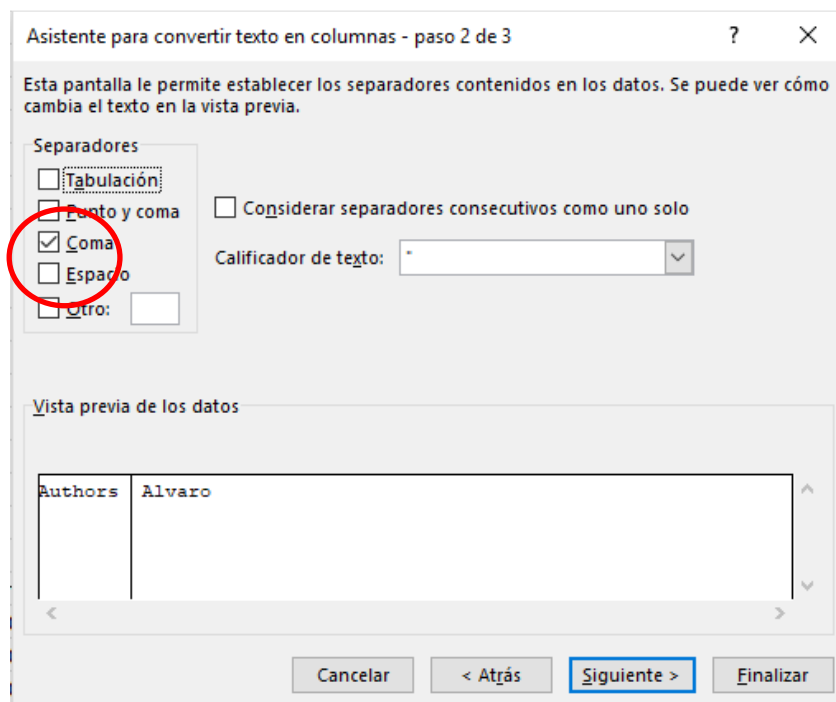


## ANEXO II: Procesado del fichero de resultados de la GUI de MATLAB

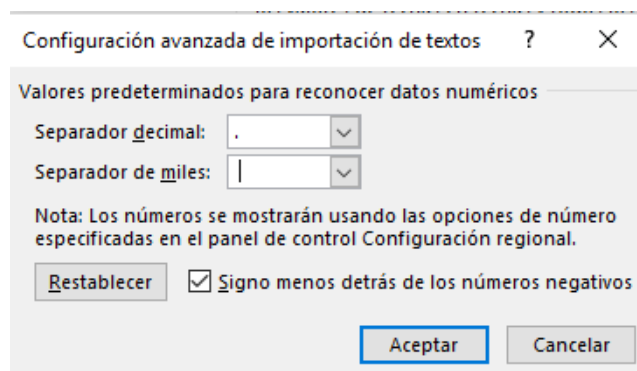
- Si seleccionamos en 'texto en columnas' se abrirá una ventana, y seleccionamos en ella la opción 'delimitados':



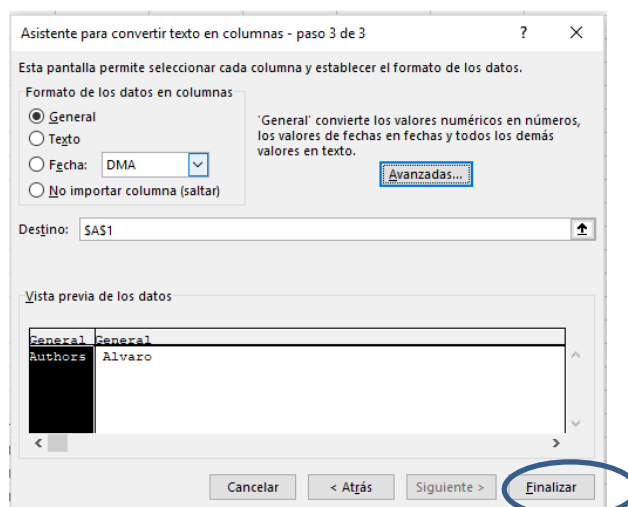
- Pulsamos en siguiente, y marcamos la opción 'coma'



- El programa interpreta que el separador de los elementos es la coma. Después, pulsamos en siguiente, y seleccionamos 'avanzadas'. Tenemos que indicar que el separador de decimal sea el punto '.', y que no queremos separador de miles. Pulsamos en aceptar.



- Pulsamos en finalizar.



## ANEXO II: Procesado del fichero de resultados de la GUI de MATLAB

- Finalmente el resultado de la representación de los datos es el siguiente:

	A	B	C	D	E	F	G	H	I
1	Authors	Alvaro							
2	Date	30-May-2019							
3	Time	12 h 22 min 44 sec							
4	Comments	No							
5	Klystron Mod	VKP-8352A							
6	Klystron S/N	2							
7	Vbeam(KV)	80							
8	Ibeam(A)	41							
9	freq(MHz)	352,21							
10	HV_pulse_dur	1,6							
11	RF_pulse_dur	1,4							
12	Repetition_f	14							
13	C1(dB)	60							
14	Slope1(V/dB	-0,025							
15	Vo1(V)	1,0792							
16	Lcable(dB)	0,7							
17	C2	83,7							
18	Slope2(V/dB	-0,0244							
19	Vo2(V)	1,0844							
20	Vbeam(V)	Ibeam(V)	Vdriver(V)	Vkly(V)	Pgen(dBm)	Pdriver(W)	Pkly(KW)	Vbeam(KV)	Ibeam(A)
21	-5,645	-2,9358	1,6936	1,2754	-10	3,5	38,6	-84,7	-29,4
22	-5,645	-2,9456	1,6494	1,2308	-9	5,2	58,9	-84,7	-29,5
23	-5,6425	-2,9638	1,6098	1,1931	-8	7,5	84	-84,6	-29,6
24	-5,6383	-2,9167	1,564	1,1553	-7	11,5	120,1	-84,6	-29,2
25	-5,6457	-2,9478	1,5278	1,1214	-6	16,1	165,4	-84,7	-29,5
26	-5,6417	-2,9258	1,4963	1,1048	-5	21,5	193,4	-84,6	-29,3
27	-5,6363	-2,9469	1,4359	1,0484	-4	37,4	329,1	-84,5	-29,5
28	-5,6463	-2,95	1,3997	1,0194	-3	52,2	433	-84,7	-29,5
29	-5,6367	-2,9467	1,3655	0,9931	-2	71,6	554,7	-84,6	-29,5
30	-5,6375	-2,935	1,3506	0,9819	-1,5	82,1	616,9	-84,6	-29,4
31	-5,6417	-2,9558	1,3342	0,9692	-1	95,5	695,5	-84,6	-29,6
32	-5,6425	-2,9581	1,3175	0,9537	-0,5	111,4	804,4	-84,6	-29,6
33	-5,6383	-2,9083	1,301	0,9387	0	129,6	926,7	-84,6	-29,1
34	-5,6367	-2,9175	1,2829	0,9241	0,5	153,2	1064,4	-84,6	-29,2
35	-5,6383	-2,9467	1,2696	0,9171	1	173,2	1136,9	-84,6	-29,5
36	-5,64	-2,9275	1,2546	0,9087	1,5	198,8	1229,9	-84,6	-29,3
37	-5,63	-2,9317	1,2409	0,9043	2	225,6	1283,3	-84,5	-29,3
38	-5,64	-2,95	1,2305	0,8975	2,5	248,3	1367,7	-84,6	-29,5
39	-5,6383	-2,9433	1,2185	0,8962	3	277,1	1383,9	-84,6	-29,4